

**UNIVERSIDAD NACIONAL DE PIURA**  
**FACULTAD DE CIENCIAS**  
**ESCUELA PROFESIONAL**  
**DE INGENIERÍA ELECTRÓNICA Y**  
**TELECOMUNICACIONES**

**TESIS:**

**PARA OPTAR EL TÍTULO PROFESIONAL DE**  
**INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES**

**BACHILLER:**

**CARLO OMAR LÓPEZ RAMÍREZ**

**ASESOR:**

**ING. EDUARDO OMAR AVILA REGALADO**

**PIURA - 2014**

## **PRESENTACION**

**Señores Miembros del Jurado:**

De conformidad con los requerimientos estipulados en el Reglamento de Grados y Títulos de la Universidad Nacional de Piura, para optar el Título de Ingeniero Electrónico y Telecomunicaciones, se pone a vuestra consideración la presente Tesis titulada:

### **“DISEÑO E IMPLEMENTACION DE UN SISTEMA DE RECONOCIMIENTO FACIAL USANDO MATLAB”**

Piura, ..... del 2014

***BACH. CARLO OMAR LÓPEZ RAMÍREZ***

# TITULO

## **“DISEÑO E IMPLEMENTACION DE UN SISTEMA DE RECONOCIMIENTO FACIAL USANDO MATLAB”**

**TESISTA:**



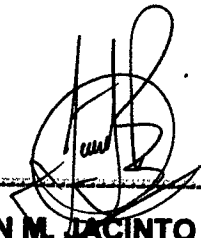
**Bach. Carlo Omar López Ramírez**

**ASESOR:**

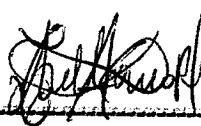


**Ing. Eduardo Omar Avila Regalado**

**JURADO CALIFICADOR:**



**Ing° JUAN M. JACINTO SANDOVAL  
PRESIDENTE**



**Ing° CARLOS E. ARELLANO RAMÍREZ  
SECRETARIO**



**Ing° OMAR E. BARRA ZAPATA  
VOCAL**

## **DEDICATORIA.**

**“A ti Dios,**

por mostrarme el camino, porque haces cosas  
maravillosas, por la luz que encamina mi sendero,  
por permitirme alcanzar uno de los sueños más  
importantes de mi vida, por enseñarme a  
levantarme de mis caídas, por los éxitos, por  
permitirme ser uno de tus hijos, por las constantes  
luchas, y por todo ...”

**A mis padres,**

Luk María Ramírez Velásquez, Severino  
López Jiménez, por el inmenso amor y por  
todo su apoyo en mi formación como  
persona.

**A mi hermano,**

Cristhian, por la perseverancia y paciencia y su  
apoyo siempre a pesar de las dificultades

A mi tía Maritza, a mis tíos. A mis primos.

A mis abuelos. A mis amigos”.

## **AGRADECIMIENTO**

- A mis padres Luk María Ramírez Velásquez, Severino López Jiménez y mi hermano Cristhian, por el inmenso apoyo brindado a lo largo de la carrera, por la perseverancia y la paciencia que tuvieron a pesar de todas las dificultades, no logre escuchar de ellos un detente, por lo contrario; siempre su meta era la de perseverar y lograr el objetivo.
- A mi familia, mi tía Maritza, a mi mamá Vicenta, y los que me rodearon, agradeciendo el inmenso apoyo, porque fueron el eje motriz principal para lograr el objetivo.
- A mi asesor y amigo, el Ing. Eduardo Omar Ávila Regalado, por la comprensión, el trabajo y la perseverancia a lo largo de todo este proyecto.
- A todos los antes mencionados, las palabras quedarían cortas para expresar todo el inmenso agradecimiento, solo queda decir, trabajo mas perseverancia es igual a resultados.

**GRACIAS.**

**CARLO OMAR LOPEZ RAMIREZ.**

## **INDICE**

<b>INTRODUCCION.....</b>	<b>1</b>
<b>CAPITULO I (PLANTEAMIENTO METODOLOGICO).....</b>	<b>3</b>
1.1 DESCRIPCION DE LA REALIDAD PROBLEMÁTICA.....	4
1.2 DEFINICION DEL PROBLEMA PRINCIPAL.....	4
1.3 FORMULACION DEL PROBLEMA.....	5
1.4 OBJETIVOS.....	5
1.4.1 OBJETIVO PRINCIPAL.....	5
1.4.2 OBJETIVOS ESPECIFICOS.....	5
1.5 HIPOTESIS.....	6
1.6 VIABILIDAD DEL ESTUDIO.....	6
1.6.1 ECONOMICA.....	6
1.6.2 TECNICA.....	6
1.6.3 OPERATIVA.....	6
1.7 JUSTIFICACION.....	6
<b>CAPITULO II (MARCO TEORICO).....</b>	<b>8</b>
2.1 BIOMETRIA.....	9
2.2 RECONOCIMIENTO BIOMETRICO.....	10
2.3 ETAPAS DEL DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO DE CARAS.....	12
2.4 TECNICAS DE RECONOCIMIENTO FACIAL 2D.....	13
2.4.1 ANALISIS DE COMPONENTES PRINCIPALES.....	13
2.4.2 ELASTIC BUNCH GRAPH MATCHING (EBGP).....	15
2.4.3 MODELO ACTIVO DE APARIENCIA (AAM).....	16
2.5 CONCEPTOS GENERALES.....	19
2.5.1 PIXEL.....	19
2.5.2 IMAGEN DIGITAL.....	19
2.5.3 CLASIFICACION DE IMÁGENES DIGITALES.....	20
2.5.4 ESPACIOS DE COLOR.....	21
2.5.4.1 MODELO RGB.....	21
2.5.4.2 MODELO HSV.....	22
2.5.4.3 MODELO YCb Cr.....	23
2.5.5 PROCESAMIENTO DE LA IMAGEN.....	24
2.5.5.1.1 TECNICAS DE MODIFICACION DEL HISTOGRAMA.....	24
2.5.5.1.2 FILTROS MORFOLOGICOS.....	28
2.5.5.1.3 FILTROS DE TEXTURA.....	33

2.5.5.1.4	FILTROS DE RECORRIDO.....	33
2.5.5.1.5	FILTRO RMS (ROOT.MEAN-SQUARE).....	33
2.5.5.1.6	OPERADORES DEL MOMENTO.....	34
2.6	DETECCION DE CARAS.....	34
2.6.1	TECNICAS BASADAS EN RASGOS.....	34
2.6.2	ANALISIS DE BAJO NIVEL.....	35
2.6.3	ANALISIS DE RASGOS.....	37
2.6.4	TECNICAS BASADAS EN LA IMAGEN.....	38
2.6.4.1	METODOS BASADOS EN SUB ESPACIOS.....	38
2.7	RECONOCIMIENTO DE CARAS.....	39
2.7.1	METODOS HOLISTICOS.....	39
2.7.1.1	ANALISIS DE COMPONENTES PRINCIPALES (PCA).....	40
2.7.1.2	ANALISIS DE COMPONENTES INDEPENDIENTES.....	44
2.7.1.3	ANALISIS DISCRIMINANTE LINEAL.....	45
2.7.1.4	METODOS BASADOS EN KERNELS.....	47
2.7.1.5	EVOLUTIONARY PURSUIT: EP.....	48
	CAPITULO III (SISTEMA, DISEÑO Y DESARROLLO).....	50
3.1	METODOLOGIA DEL PROYECTO.....	51
3.1.1	APLICACIÓN DE LOS CONCEPTOS DE RECONOCIMIENTO FACIAL..	51
3.1.2	DESARROLLO DE LA DETECCION Y MARCADO DE PATRONES BIOMETRICOS.....	51
3.1.3	DESARROLLO DE LA IDENTIFICACION BIOMETRICA.....	51
3.2	ESQUEMA DEL SISTEMA.....	51
3.2.1	SISTEMAS BIOMETRICOS.....	51
3.2.2	MODO DE IDENTIFICACION.....	53
3.2.3	DESCRIPCION DEL SISTEMA DESARROLLADO.....	55
3.3	DISEÑO DEL PROGRAMA DE RECONOCIMIENTO E IDENTIFICACION..	56
3.3.1	DISEÑO DEL SOFTWARE PCA.....	56
3.3.2	DISEÑO DE INTERFAZ HMI.....	56
3.3.3	ALGORITMO Y CODIGO DEL PROGRAMA PCA.....	57

3.3.3.1 RUTA DE BASE DE DATOS.....	57
3.3.3.2 CONFIGURACION Y DETECCION DE LA WEBCAM.....	58
3.3.3.3 CAPTURA Y ALMACENAMIENTO DE LA IMAGEN.....	59
3.3.3.4 PROCESAMIENTO DEL ARCHIVO INPUT.JPG.....	60
A. VARIABLES COORDENADAS.....	60
B. DETECCION DEL ROSTRO.....	61
C. RECORTE DEL ROSTRO.....	63
D. EXTRACCION DE LOS OJOS.....	64
E. RECORTE DE LA IMAGEN DE LOS OJOS.....	69
F. ROTACION DE LA IMAGEN DEL ROSTRO.....	72
G. DETERMINACION DE LOS DEMAS PUNTOS DE LOS OJOS.....	73
H. OBTENIENDO RATIOS.....	75
 CAPITULO IV.....	 76
4.1 PRUEBAS REALIZADAS.....	77
4.1.1 VALIDEZ DE IMAGEN DE ROSTRO HUMANO.....	77
4.1.2 IMAGEN DE ROSTRO INCOMPLETO.....	78
4.1.3 AREA DE LA MASCARA DE ROSTRO NO VALIDA.....	78
4.1.4 DISTANCIA MEDIDA A 70 CM.....	79
4.2 CONCLUSIONES.....	79
 BIBLIOGRAFIA.....	 81



## INTRODUCCION

En la actualidad el Perú es un país con alto índice de delincuencia por lo que se ha llegado a situaciones extremas instalando sistemas de seguridad que permita de alguna manera dar soluciones ante hechos delictivos. En el campo de la seguridad, los usuarios deben recordar información (claves) y poseer tarjetas de acceso. Las claves pueden ser olvidadas, las tarjetas pérdidas o robadas. Ambos elementos requieren un esfuerzo consciente del participante para que funcionen. La biometría en cambio, usa características específicas e intransferibles de los usuarios, difíciles de duplicar o alterar y con menor esfuerzo de uso.

Existen varios tipos de biometría: huellas digitales, geometría de la mano, retina, rostro, firma, voz, etc. De todo el grupo, el rostro es el menos invasivo pues puede procesarse incluso sin que éste tenga conocimiento de ello.

La investigación para autenticar por medio del rostro surge desde la década de los 70 con muchas limitaciones por la cual la tecnología no era la más sofisticada, desde ahí se dejó de lado hasta la década de los noventa donde comienzan a investigar y hay algunos resultados con tecnología mucha más avanzada, y hasta la actualidad es un tema de investigación.

Los motivos que despertaron el interés de esta Tesis como *trabajo de investigación* es la investigación y el uso de las herramientas Matlab y los algoritmos de reconocimiento biométrico para el desarrollo de aplicaciones de seguridad que hoy se requiere en la actualidad como por ejemplo para su uso como sistemas de seguridad en centros comerciales, centros de recreación, etc. Es por ello que en los últimos años se ha explotado los sistemas biométricos como métodos efectivos de identificación de personas a través de características físicas únicas como distancia entre los ojos, anchura de la nariz, forma de la barbilla, pómulos, forma de la boca, etc.

El objetivo planteado en la Tesis es *diseñar e implementar una aplicación de reconocimiento automático de rostros humanos utilizando la herramienta matemática Matlab y una cámara WEBCAM para la captura de imágenes.*

El desarrollo de la Tesis se basa en cuatro capítulos: en el **Capítulo I** se desarrolla el planteamiento metodológico donde se describe brevemente la realidad problemática y el problema principal a enfrentar, el **Capítulo II** consta del marco teórico y que es la base fundamental para el desarrollo de la Tesis como lo es el método PCA o análisis de componentes principales para el reconocimiento de rostros humanos, en el **Capítulo III** se describe el diseño y desarrollo de la aplicación desarrollada en el Matlab; aquí se hace énfasis a los algoritmos principales que permite hallar los parámetros principales de comparación para el reconocimiento biométrico, y en el **Capítulo IV** se muestra las pruebas realizadas al software de reconocimiento para validar hasta que distancia se puede enfocar la WebCam para tener un reconocimiento del rostro. Finalmente se plantea las **Conclusiones** de la investigación además de indicar las **Referencias Bibliográficas** usadas para el desarrollo de la Tesis.

*El autor*

# **CAPITULO I**

# **PLANTEAMIENTO METODOLOGICO**

**TESIS PARA OBTENER EL TITULO PROFESIONAL  
INGENIERO ELECTRONICO Y  
TELECOMUNICACIONES  
UNIVERSIDAD NACIONAL DE PIURA**

## 1.1 DESCRIPCION DE LA REALIDAD PROBLEMÁTICA

Cada vez hablando en el contexto general se diseñan nuevas formas de seguridad y a la vez se crean tecnologías cada vez más sofisticadas, todas estas tecnologías y nuevos diseños solo tienen un fin en común, SEGURIDAD. Pero cuando hablamos de seguridad es en un contexto muy generalizado, existen diferentes métodos de seguridad, como por ejemplo, alarmas, sistemas de seguridad con ubicación satelital GPS, el uso de cámaras de seguridad, (cámaras de seguridad convencional, cámaras de seguridad IP, cámaras de seguridad con infrarrojo, etc.), cuya finalidad es ser colocados como elementos disuasivos ante hechos delictivos. Cada vez que ingresa al mercado nuevos sistemas de seguridad más sofisticados, el concepto se hace cada vez más complicado, aunque su finalidad sigue siendo la misma, debido a que su uso está orientado a anticipar o prevenirnos ante cualquier peligro al que nos exponemos.

Cuando hablamos de seguridad utilizando cámaras de video, no solo se trata de convencernos de que su finalidad es prevenir, debido a que solo grabará un video porque tendrá incluido un DBR y un monitor para captar la imagen de la zona que queremos visualizar, y que una vez visualizada la imagen ahí termino toda la información que nos brinda la cámara la pregunta es *¿que sabemos del sujeto que está ahí en la imagen?* sabremos acaso si viene sin arma porque no la visualizamos; en fin la idea de un trabajo de seguridad es que además de visualizar la imagen esta nos proporcione información y esto es lo que proponemos en este trabajo de investigación.

## 1.2 DEFINICION DEL PROBLEMA PRINCIPAL

La descripción de la realidad problemática en temas de seguridad es amplia, debido a que se han desarrollado diversos sistemas de seguridad con diversas tecnologías, pero en esta sesión se planteara el problema concreto sobre la cual versa la investigación. Y el problema principal se orienta a que *la imagen de una persona que se muestra en una Webcam no me permite identificar a la persona si esta no está asociada a un algoritmo de reconocimiento de rostros* es por ello que no se sabrá si se cometerá un acto delictivo.

### **1.3 FORMULACION DEL PROBLEMA**

Definido el problema, se formula la siguiente pregunta de investigación:

*¿Es posible el diseño e implementación de una aplicación en base la herramienta matemática Matlab y la aplicación de algoritmos biométricos para el reconocimiento de rostros?*

### **1.4 OBJETIVOS.**

#### **1.4.1 OBJETIVO PRINCIPAL**

- Diseñar e implementar una aplicación para el reconocimiento automático de rostros humanos en condiciones adecuadas en su componente de software e interfaces para dispositivos de hardware.

#### **1.4.2 OBJETIVOS ESPECÍFICOS**

- Desarrollar un sistema de reconocimiento de rostros en base a los algoritmos de reconocimiento biométrico.
- Desarrollar un prototipo del sistema de software que integra los módulos de captura, procesamiento, almacenamiento e interfaz con el hardware de control.
- Estudiar las técnicas de procesamiento digital de imágenes que permitan minimizar tasa de error en el reconocimiento de rostros.
- Validar el sistema empleando los métodos convencionales descritos en la literatura científica especializada.
- Investigar sobre algoritmos de comparación de imágenes disponibles en el medio académico y comercial.
- Diseñar e implementar un modelo de datos que soporte el almacenamiento y búsqueda imágenes digitales de rostros humanos, así como la forma canónica de dichas imágenes.
- Diseñar e implementar el software de captura de imágenes de rostros mediante una cámara web, procesarla y almacenarla en la base de datos de imágenes.

## **1.5 HIPÓTESIS**

*El conocimiento de herramientas de desarrollo para el procesamiento de imágenes de Matlab y los algoritmos de reconocimiento biométrico posibilita el diseño e implementación de una aplicación para el reconocimiento de rostros humanos*

## **1.6 VIABILIDAD DEL ESTUDIO**

- 1.6.1 Económica:** Se cuenta con la herramienta matemática Matlab y el presupuesto necesario para tener el servicio de Internet y otros gastos que permitirá ser el soporte fundamental para el desarrollo de la aplicación.
- 1.6.2 Técnica:** Se adquirió una WebCam con la resolución necesaria (10MPixel) para la obtención de las imágenes que serán procesadas por el sistema de reconocimiento facial, así como una computadora HP Z600 para el desarrollo de la aplicación.
- 1.6.3 Operativa:** El conocimiento de temas en procesamiento de imágenes desarrollado en la escuela de Ingeniería Electrónica así como la elaboración de algoritmos de procesamiento con Matlab, así como el apoyo del asesor de la tesis permitirá el desarrollo de la Tesis.

## **1.7 JUSTIFICACION**

El concepto de la inseguridad ciudadana ante hechos delictivos que se suceden día a día en el país, ya sea la propiedad pública o privada o las mismas personas; permitió que se tomen todo tipo de medidas de prevención, pero el problema que aún no se resuelve es identificar plenamente a los delincuentes que actúan fuera de la ley. La tecnología emergente en cuanto al desarrollo de video cámaras para la captura de imágenes en tiempo real, así como herramientas matemáticas que se han desarrollado para el reconocimiento biométrico, sirven como eje principal para plantear el desarrollo de dispositivos inteligentes que, contando con una base de datos de rostros, nos permita minimizar la inseguridad, usando esta tecnología como elementos disuasivos más potentes que las simples cámaras de video vigilancia.

**Por lo expuesto, la aplicación de la Tesis queda justificada, porque este estudio constituye en una herramienta necesaria para los actores involucrados en el proceso de mejora de la seguridad ciudadana, y en forma indirecta, para los responsables de la seguridad de establecimientos; porque de su correcta aplicación se obtienen beneficios tangibles en el rendimiento y mejora de la calidad de vida.**

# **CAPITULO II**

# **MARCO TEORICO**

**TESIS PARA OBTENER EL TITULO PROFESIONAL  
INGENIERO ELECTRONICO Y TELECOMUNICACIONES  
UNIVERSIDAD NACIONAL DE PIURA**



*"Nunca olvido una cara pero con la suya voy a hacer una excepción."*

*Groucho Marx*

## **2.1 BIOMETRIA**

***La biometría es la ciencia que se dedica a la identificación de individuos a partir de una característica anatómica o un rasgo de su comportamiento.*** Busca obtener, clasificar y utilizar la información de esas características (anatómicas o de comportamiento) para reconocer e identificar a las personas, restringir el acceso a sitios no permitidos, controlar horarios en empresas, autenticar información, y muchas otras aplicaciones. Para esto utiliza equipos electrónicos que desarrollan las mediciones biométricas, y algoritmos que permiten digitalizar, clasificar y almacenar la información para poder utilizar después.

No cualquier característica anatómica puede ser utilizada con éxito por un sistema biométrico. Para que eso sea así debe cumplir con las siguientes características: universalidad, unicidad, permanencia y cuantificación.

A continuación se muestra una tabla en la que se recogen las diferentes características de algunos sistemas biométricos.

**Tabla 1.1 Características de los sistemas biométricos**

	Ojo (Iris)	Ojo (retina)	Huellas dactilares	Geometría de la mano	Escritura y firma	voz	Cara
fiabilidad	Muy alta	Muy alta	alta	alta	Media	Alta	Alta
Facilidad de uso	Media	Baja	Alta	Alta	Alta	Alta	Alta
Prevención de ataques	Muy alta	Muy alta	Alta	Alta	Media	Media	Media
Aceptación	Media	Media	Alta	Alta	Muy alta	Alta	Muy alta
estabilidad	Alta	Alta	Alta	Media	Baja	Media	Media

## **2.2 RECONOCIMIENTO BIOMÉTRICO**

El reconocimiento biométrico se define como el reconocimiento automático de individuos basado en sus características fisiológicas o de comportamiento.

Las huellas dactilares, las retinas, el iris, los patrones faciales de venas de la mano o la geometría de la palma de la mano, representan ejemplos de características físicas (estáticas), mientras que entre los ejemplos de características del comportamiento se incluye la firma, el paso y el tecleo (dinámicas). La voz se considera una mezcla de características físicas y del comportamiento, pero todos los rasgos biométricos comparten aspectos físicos y del comportamiento.

Las características deben ser invariables en el tiempo, representadas a través de una medición numérica, distinguibles entre una persona y otra, y universales, es decir que cada persona la tenga. Adicionalmente, los sistemas biométricos deben contar con una serie de requisitos: deben ser seguros de tal forma que no puedan ser burlados usando métodos fraudulentos; deben tener un alto rendimiento trabajando a velocidades razonables, con bajas tasas de error; deben tener una alta aceptabilidad por los usuarios ya que voluntariamente ellos deben aceptar someterse a ser medidos.

La mayoría de los sistemas biométricos utilizados en zonas de seguridad suelen estar basado en huellas dactilares, pero este es un método muy intrusivo, es decir, obliga a que el usuario sea muy colaborativo. No es posible tomar la huella de una persona sin que se percate de ello y por lo tanto, de su consentimiento.

Sin embargo un sistema biométrico basado en rasgos faciales permite el tratamiento de imágenes sin que los sujetos estudiados se percaten de ello.

Además, los sistemas basados en caras son muy útiles a los operadores humanos, ya que utilizan imágenes que luego pueden ser mostradas para su cotejo por parte de dicho operador.

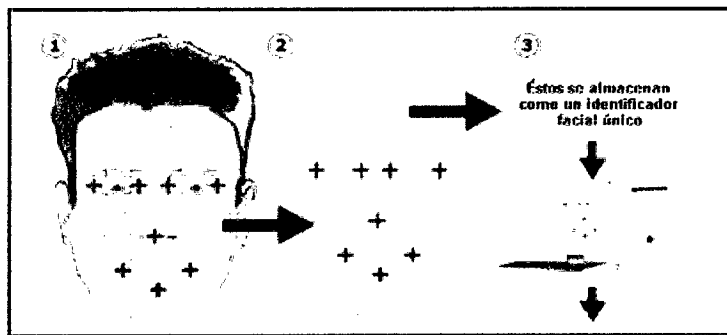


Fig. 2.1 Secuencia Digital del Reconocimiento Biométrico

De todos los rasgos anatómicos, el rostro es el elemento que con más frecuencia utilizamos los seres humanos para identificar a otro individuo. Para ello, el cerebro comienza por establecer los aspectos físicos de una cara, a continuación determina si estas acciones son conocidas o no y, y por último, procede a otorgar un nombre a lo que ve.

Este proceso tan aparentemente sencillo para nosotros puede resultar muy difícil para una máquina. Además, nuestra habilidad es tal que somos capaces de reconocer la fisonomía de cualquier persona obviando detalles como la longitud del pelo, ciertos complementos (gafas, sombreros, etc.) o incluso los cambios que experimenta el rostro por el paso del tiempo. En la figura 2.1 se muestra un ejemplo de la secuencia digital del reconocimiento biométrico la cual será realizada por un sistema biométrico el cual fundamenta sus decisiones de reconocimiento mediante una característica personal que puede ser reconocida o verificada de manera automatizada. Este sistema de reconocimiento facial, lo primero que realiza es la detección de los ojos de los usuarios y después se centra en los rasgos situados bajos los ojos y encima de los labios. La información relevante es transferida a una base de datos en forma de algoritmo numérico el cual genera un identificador facial único para cada usuario.

## 2.3 ETAPAS DEL DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO DE CARAS

Tabla 1.2: Escenarios de aplicación del reconocimiento de caras

<b>CATEGORÍA</b>	<b>ESCENARIO DE EJEMPLO</b>
Identificación de cara	Carnés de conducir, programas de derecho, inmigración, pasaportes, registro de votos, registro de la seguridad social.
Control de acceso	Acceso de vehículos. Facilidad de acceso, acceso a ordenadores, acceso a programas de ordenador, acceso a redes de ordenadores, acceso a programas online.
Seguridad	Alerta terrorista, sistemas de seguridad a bordo, archivos médicos, seguridad de ordenadores, seguridad en internet.
Vigilancia	Vigilancia en plantas nucleares, vigilancia en parking, vigilancia en los vecindarios
<b>Smarcards</b>	<b>Autenticación de usuario</b>
Aplicaciones en la ley	Reconocimiento de criminales, identificación de estafadores en casinos, fraude a la seguridad social.
Bases de datos	Catalogación y recuperación de caras, etiquetado automático de caras, clasificación de caras.
Gestión multimedia	Búsqueda en bases de caras, resumen y segmentación de caras basadas en video.
<b>Interacción hombre-máquina</b>	<b>Juegos interactivos</b>
Otros	Verificación de fotos antiguas, transmisión de imágenes de baja tasa de bit.

## **2.4 TÉCNICAS DE RECONOCIMIENTO FACIAL 2D**

Existen desarrollos de muchos métodos de reconocimiento facial, aunque la mayoría de estos métodos están basados en imágenes 2D: como por ejemplo los modelos denominados eigenfaces (son un conjunto de vectores propios utilizados en el problema de la visión por ordenador de reconocimiento de la cara humana [1]), fisherfaces (La misma persona con la misma expresión facial y vista desde el mismo punto de vista puede aparecer dramáticamente diferente cuando las fuentes de luz iluminan la cara desde diferentes direcciones[2]), active appearances, ente otras.

Las dos técnicas más conocidas son: PCA (Principal Components Analysis) y LDA (Linear Discriminant Analysis, también conocida como Fisher Discriminant Analysis, FDA).

PCA es el más utilizado en la técnica de reconocimiento conocida como autocaras (Eigenfaces) mientras que FDA o LDA es utilizado en Fisherfaces.

### **2.4.1 ANALISIS DE COMPONENTES PRINCIPALES (PCA)**

El método más utilizado y es en el que se basa el desarrollo de este proyecto de tesis, es reconocido por su simplicidad además que proporciona buenos resultados, consiste en proyectar las imágenes en un espacio unidimensional.

Se transforma la imagen [ $n \times m$  píxeles] en un vector de dimensiones [ $n.m$ ]. Este espacio  $n.m$  es demasiado grande como para permitir un reconocimiento de objetos en forma rápida y robusto, para ello se ha hecho uso de la técnica PCA.

El análisis PCA deriva de la transformada de Karhunen-Loewe (KL). Dada una representación vectorial  $S$ -dimensional de cada cara, en un conjunto de imágenes de entrenamiento, el análisis PCA tiende a encontrar un subespacio  $t$ -dimensional ( $t < s$ ) cuya base de vectores corresponde a las direcciones de máxima varianza del espacio de la imagen original. Este nuevo subespacio es normalmente de dimensión inferior ( $t \ll s$ ). La

base de vectores del PCA es definida como auto vectores de la matriz de covarianza (matriz de dispersión).

Un autovector cuya dimensión es la misma que las imágenes iniciales y por lo tanto se puede ver como una imagen en el llamado espacio imagen. Al hacerlo así, da la apariencia de imágenes de caras, pero muy diferentes del conjunto inicial. Por esta razón, en la literatura se conocen como autocaras (eigenfaces). La figura 2.2 muestra los autovectores de un conjunto de entrenamiento.

Los primeros auto vectores representan las direcciones de mayor varianza en la región de las caras. Cada valor de un autovector es una coordenada en el espacio imagen, representando la contribución de ese dato a la varianza de la región de caras en esa dirección. Un valor grande de esa cifra representa una contribución.

Importante como las imágenes se muestran en blanco y negro, valores muy próximos al blanco o al negro representan contribuciones significativas de esas magnitudes.



Fig. 2.2 Eigenfaces de varios rostros

### 2.4.2 METODO BASADO EN ONDICULAS (EBGP)

Los métodos basados en ondiculas (wavelets), utilizan esta transformación para identificar los bordes de las características faciales. El análisis EBGP (Elastic Bunch Graph Matching) parte de la idea de que todas las caras comparten una estructura topológica similar.

El algoritmo EBGM tal como se muestra en la figura 2.3 construye rasgos locales (considerando solo la región próxima a un punto dado), extraídos a partir del uso de transformada wavelet de Gabor y una figura global de la cara representada como un grafo.

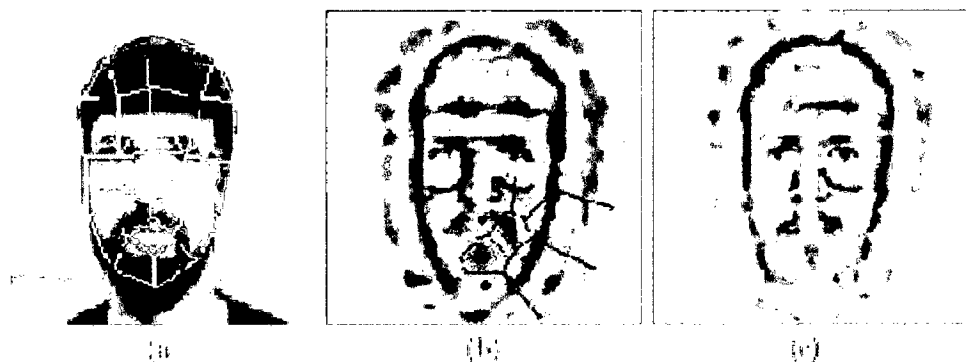


Fig. 2.3 Representación interna de la aproximación EBGM (a) representa un grafo sobre una cara. (b) es la reconstrucción de la cara a partir del grafo. (c) representa la reconstrucción de la cara a partir de un grafo conjunto en el que se ha considerado en cada nodo el jet que mejor se adapta.

Las caras son representadas como grafos, con nodos situados en puntos de la cara, llamados “fiducial points” (ojos, nariz, etc.); y por bordes, etiquetados por vectores de distancia. Cada nodo contiene un conjunto de 40 coeficientes complejos de la transformación wavelet de Gabor (5 frecuencias x 8 orientaciones), a diferentes escalas y orientaciones (fase, amplitud). A estos coeficientes les denominan “jets”. esto se muestra en la figura 2.4

El reconocimiento está basado en gráficos etiquetados (face graph). Un gráfico etiquetado es un conjunto de nodos conectados por bordes, los nodos están etiquetados por jets y los bordes etiquetados por distancias.

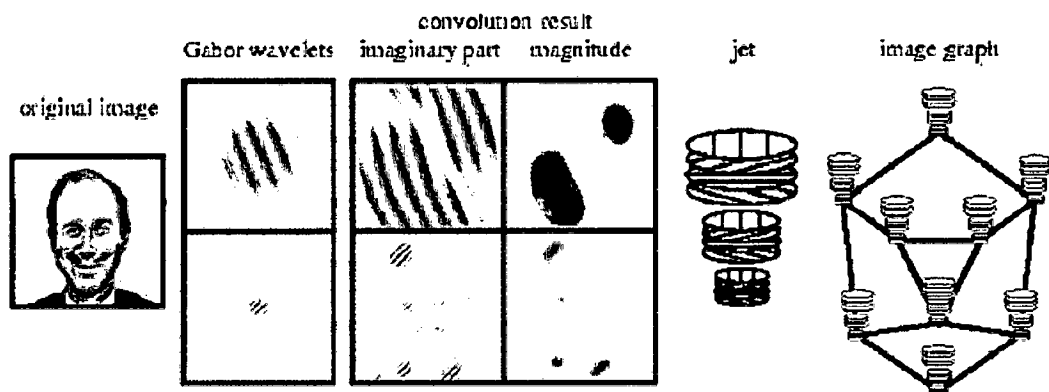


Fig.2.4 para aclarar la definición de jet, solo se muestran en la figura, 3 frecuencias y 4 orientaciones. Una colección de este jets, junto con información sobre su localización constituye un imagen graph, usado para representar un como objeto, como por ejemplo, una cara.

#### 2.4.3 MODELO ACTIVO DE APARIENCIA (AAM)

El problema clave de un modelo de cara es la variabilidad y obviamente, para que un modelo sea útil y valido necesita ser específico, es decir, que sea capaz de representar solo ejemplos validos del tipo de objeto que hemos modelado. Estos modelos deben ser capaces de representar la variabilidad, así que tienen que ser modelos deformables.

Un modelo de este tipo es el AAM (Active Appearance Models).

Estos modelos se construyen a partir del análisis estadístico de la estructura interna de un objeto cuyo contorno está marcado por puntos, estos puntos reciben el nombre de LANDMARKS (puntos de referencia).

AAM, es un modelo estadístico integrado que combina un modelo de variaciones de formas con un modelo de variaciones de apariencia. El modelo estadístico, en el que las formas y variaciones del nivel de gris del objetivo de interés, pueden generalizarse para la mayoría de los objetos.

Aplicando a una imagen, significa encontrar los parámetros del modelo que generan una imagen sintética lo más cerca posible a la imagen objetivo.



Los tres pasos para generar el modelo en el sistema AAM son:

**CAPTURA → NORMALIZACIÓN → ANÁLISIS ESTADÍSTICO**

Estos tres pasos se aplican a las distintas imágenes que tengamos en nuestro conjunto de entrenamiento.

**a. Captura:**

*Forma:* se define un número determinado de puntos sobre el contorno del objeto que los interesa para determinar la forma (ver figura 2.5).



Fig. 2.5 Rostro con LandMarks

*Textura:* se muestra una imagen usando antes una función de ajuste (Warping función) como piece-wiseaffine [19], thinplate [20]... y separando, en zonas a comparar, la textura mediante un esquema de triangulación.

**b. Normalización:**

*Forma:*se busca en espacio (escenario) normalización, para ello se alinean las formas eliminando la posición, escalado y orientación.

*Textura:*se quita de la imagen los efectos de la iluminación, etc, normalizándola, buscando un contraste y brillo estándar para todas las imágenes.

**c. Análisis Estadístico:**

*Forma y Textura:* en ambos casos se utiliza PCA para conseguir una descripción compacta y restrictiva, obteniéndose  $\mathbf{b}_s$ , vector de dimensión  $2n$ , diferente para cada una de las caras de nuestro conjunto de entrenamiento.

Para generar un modelo completo AAM debe considerarse también la textura.

Con la forma, la adquisición de los datos era directa porque los landmarks de la forma eran los propios datos. En el caso de la textura se necesita un método consistente para coger la información de textura que hay entre los landmarks, es decir se necesita dividir las imágenes en zonas iguales que nos permitan comparar los valores de textura de distintas caras de forma ordenada en todo el conjunto de entrenamiento.

Cada una de las formas de cada imagen es ajustada a una forma de referencia (Figura 2.6) que, generalmente, es la forma media y luego es muestreada, con ello se consigue, para cada imagen, un trozo de textura con el mismo número de puntos, y ya es posible procesarla.



Fig.2.6 Partición de la textura en zonas

Llegando a este punto, se realiza una normalización de valores y se aplica sobre la textura PCA, obteniendo un parámetro  $b_G$ .

El sistema decisor compara los parámetros  $b_s$  y  $b_G$  de la imagen capturada con la foto con los parámetros almacenados.

## 2.5 CONCEPTOS GENERALES

El proyecto está enmarcado dentro del campo de la visión computacional, por lo que se van a introducir algunos de los elementos básicos de la representación de imágenes.

### 2.5.1 Pixel

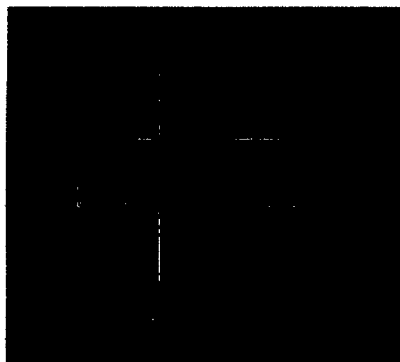
Es la abreviatura de las palabras inglesas “picture element”. Es el menor de los elementos de una imagen al que se puede aplicar individualmente un color o una intensidad o que se puede diferenciar de los otros mediante un determinado procedimiento.



Fig. 2.7 Representación de un píxel

### 2.5.2 Imagen Digital

Una imagen digital se compone de una agrupación de píxeles, cada uno con un valor de intensidad o brillo asociado. Una imagen digital se representa mediante una matriz bidimensional, de forma que cada elemento de la matriz se corresponde con cada píxel en la imagen (ver figura 2.8)



a)

21	41	45	41	40	41	41	40	41	41
23	40	41	41	40	41	41	40	41	41
21	41	41	41	200	210	210	40	40	41
21	40	41	41	200	210	210	40	40	41
21	41	21	21	200	210	210	210	41	41
21	21	21	21	200	210	210	210	210	210
41	40	41	41	200	210	210	40	41	41
21	41	41	41	200	210	210	40	40	41
21	21	21	21	40	41	41	40	41	41
21	21	41	41	40	41	41	40	41	41

b)

Fig. 2.8 a) Imagen original; b) Estructura matricial de la imagen.

### 2.5.3 Clasificación de imágenes digitales

Dependiendo del rango de los valores que puede tomar cada pixel podemos distinguir los siguientes tipos de imágenes:

- a. **Imágenes binarias:** el rango está formado por los valores negro o blanco [0 1] únicamente. En la figura 2.9 se muestra una imagen de este tipo.

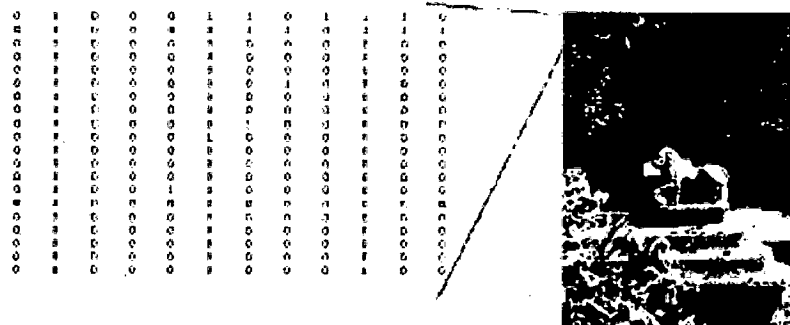


Fig. 2.9 Imagen binaria

- b. **Imagen de intensidad:** también conocidas como imágenes en escala de grises. Existen hasta 256 niveles de grises, por lo que su rango se encuentra entre [0,255].

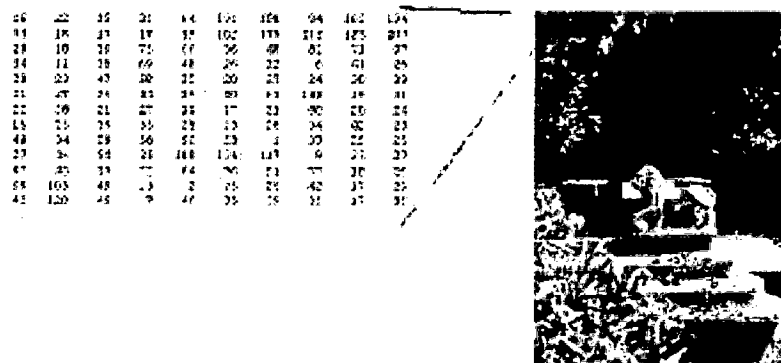
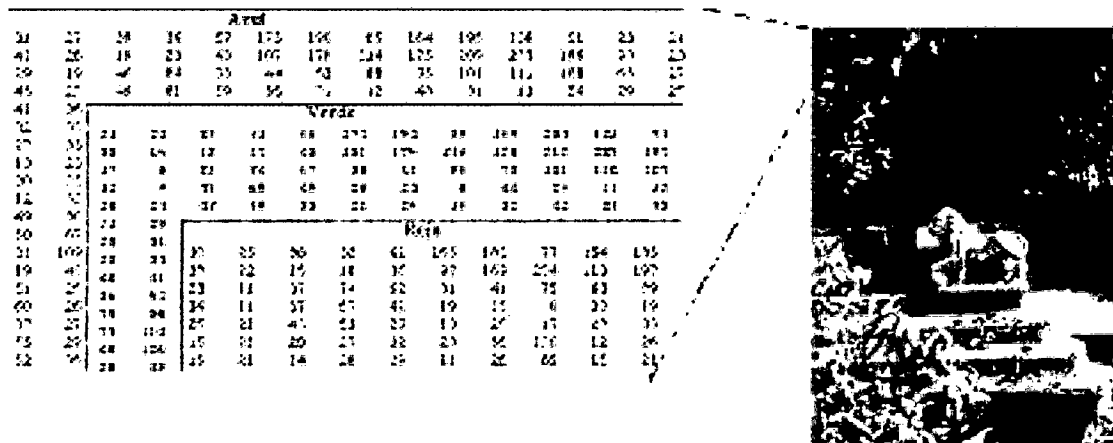


Fig. 2.10: Imagen en escalas de grises.

- c. **Imágenes en color:** todo color se puede componer a partir de tres componentes básicas. El contenido de cada pixel de la imagen es una terna de valores, un valor por cada componente de color básico.



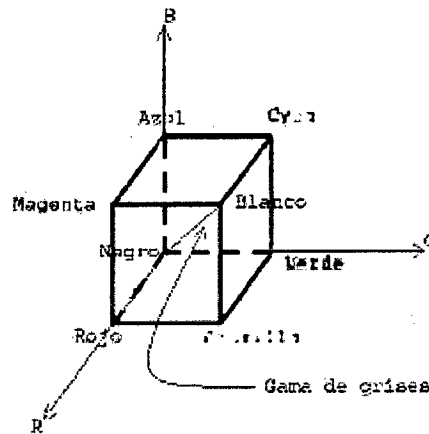


Fig. 2.12 Representación grafica del modelo de color RGB

Las imágenes con modelo RGB contienen tres planos de imágenes independientes, uno para cada color primario. El procesamiento de imágenes en color, utilizando el color RGB, toma sentido cuando las imágenes se expresan naturalmente en términos de estos tres planos.

Actualmente muchas cámaras a color utilizadas para adquirir imágenes digitales, utilizan el formato RGB. Esto convierte a modelo RGB en un modelo de gran importancia para el procesamiento de imágenes.

#### 2.5.4.2 Modelos HSV

Las siglas H, S y V corresponden a tono, saturación y valor respectivamente. También se denomina HSB, siendo B el brillo (brighness).

El sistema coordenado es cilíndrico, y el subconjunto de este espacio donde se define el color es una pirámide de base hexagonal.

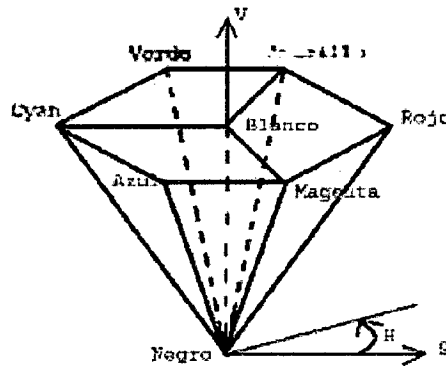


Fig. 2.13 Representación gráfica del modelo de color HSV.

El área hexagonal corresponde a un valor de  $V = 1$ , conteniendo los colores más brillantes. El tono se mide como el ángulo alrededor del eje S. El rojo se sitúa a  $0^\circ$ , el verde a los  $120^\circ$  y el azul a los  $240^\circ$ . Los colores complementarios son aquellos que se encuentren a los  $180^\circ$  del señalado. El rango de S se extiende desde 0 (coincidiendo con el eje de la pirámide) hasta 1, coincidiendo con el final del área hexagonal de la pirámide.

La obtención de este espacio de color a partir del RGB se obtiene aplicando las siguientes ecuaciones para H, S y V respectivamente:

$$H = \arccos \frac{\frac{1}{2}((R - G) + (R - B))}{\sqrt{((R - G)^2 + (R - B)(G - B))}}$$

$$S = 1 - 3 \frac{\min(R, G, B)}{R + G + B}$$

$$V = \frac{1}{3}(R + G + B)$$

Donde R, G y B son los valores del canal rojo, verde y azul respectivamente.

#### 2.5.4.3 Modelo YCbCr

YCbCr es una codificación no lineal del espacio de color RGB, usado comúnmente en la compresión de imágenes. El color es representado por la iluminancia (Y) y por dos valores diferentes de color (Cb y Cr) que son características colorimétricas del color.

El parámetro Y indica la luminosidad o la claridad del color (que se pueden ver como un tono de gris), los parámetros Cb y Cr indica la ubicación del color entre el rojo y el verde.

La obtención de este espacio de color a partir del RGB es la siguiente:

$$Y = 0.299R + 0.587G + 0.114B$$

$$Cr = R - Y$$

$$Cb = B - Y$$

Siendo R, G y B son los valores del canal rojo, verde y azul respectivamente.

La sencillez de la transformación y la separación explícita de las componentes de iluminancia y de crominancia del color, hacen a este espacio de color un método atractivo para la modelización del color de la piel.

### **2.5.5 Procesamiento de la Imagen**

El principal objetivo de las técnicas de mejoramiento de imagen es procesar una imagen con el fin de hacerla más adecuada para una determinada aplicación o procesamiento posterior. Depende por tanto del problema específico a resolver el que se emplee una u otra técnica. Los métodos de mejora de imagen se pueden dividir en dos campos diferentes:

- Métodos en el dominio frecuencial
- Métodos en el dominio espacial

Los primeros se basan en modificar la transformada de Fourier de la imagen, mientras que los segundos se basan en manipulaciones directas sobre los píxeles de la imagen.

#### **2.5.5.1 Técnicas de modificación del Histograma**

Estas técnicas van principalmente enfocadas a mejorar la visualización de una imagen. El histograma de una imagen es un gráfico que ofrece una descripción global de la apariencia de la imagen. En el eje de abscisas se representa el rango de valores de píxeles de la



imagen, mientras que en el eje de ordenadas se representa el rango de valores que pueden tomar esos píxeles. La expansión del contraste es una de estas técnicas. Consiste en que, dado un rango de valores de grises ( $ND_{max} - ND_{min}$ ) menor que el rango disponible por el por el dispositivo de visualización ( $NV_{max} - NV_{min}$ ), se está perdiendo contraste (entendido éste como relación entre los valores máximo y mínimo de una imagen). Visualmente es claro el efecto, al observar que no existe mucha diferencia entre los tonos más claros y más oscuros. Mediante distintas operaciones matemáticas se pueden transformar esos valores de grises en otros con un rango mayor que se adapte plenamente a la capacidad del dispositivo de visualización:

- a. Estiramiento Lineal:** Es la forma más sencilla de efectuar el contraste. Consiste en buscar una función lineal que ajuste de forma que el rango  $ND_{min}$  a  $ND_{max}$  se transforme en  $NV_{min}$  a  $NV_{max}$ , por lo tanto  $ND_{max} = NV_{max}$  y  $ND_{min} = NV_{min}$ . El resto de valores  $ND$  serán transformados en otros según esa transformación lineal. De forma general se puede establecer:

$$NV = a + b \cdot ND$$

Donde  $a$  es un offset y  $b$  una ganancia. Y como se conocen los valores de dos puntos:

$$NV_{min} = a + b \cdot ND_{min} ; NV_{max} = a + b \cdot ND_{max}$$

Por lo tanto los coeficientes de la transformación quedan:

$$b = \frac{NV_{max}}{ND_{max} - ND_{min}} ; a = \frac{-NV_{max}ND_{min}}{ND_{max} - ND_{min}}$$

O expresando la transformación en un solo termino:

$$NV = \frac{ND - ND_{min}}{ND_{max} - ND_{min}} NV_{max}$$

- b. Filtrado Espacial:** El filtrado espacial es la operación que se aplica a las imágenes para mejorar o suprimir detalles espaciales con el fin de mejorar la interpretación visual. Ejemplos comunes incluyen aplicar filtros para mejorar los detalles de bordes en imágenes, o para reducir o eliminar patrones de ruido. El filtrado espacial es una operación “local” en procesamiento de imagen en el sentido de que modifica el valor de cada pixel de acuerdo con los valores de los pixeles que lo rodean; se trata de transformar los ND originales de tal forma que se parezcan a diferentes más de los correspondientes a los pixeles cercanos.

La frecuencia espacial define la magnitud de cambios de los datos por unidad de distancia en una determinada zona de la imagen. Áreas de la imagen con pequeños cambios o con transiciones graduales en los valores de los datos se denominan áreas de bajas frecuencias. Áreas de grandes cambios o rápidas transiciones se conocen como áreas de altas frecuencias. Así, los filtros especiales se pueden dividir en tres categorías:

- c. Filtro paso Bajo:** Enfatizan las bajas frecuencias, suavizando las imágenes y suprimiendo ruido. Se trata de asemejar el ND de cada pixel al ND de los pixeles vecinos, reduciendo la variabilidad espacial de la imagen. Ello produce un emborronamiento de los bordes, perdiéndose en nitidez visual de la imagen, pero ganando en homogeneidad.

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Fig. 2.14: Ejemplo filtro paso bajo: Kernel

Vemos que lo que se realiza es una media aritmética de los nueve pixeles que componen la ventana de filtrado, con lo que se reducen los espurios y la variabilidad de la imagen.



Fig. 2.15 a) Imagen original; b) Imagen filtrada paso bajo con el filtro de la ilustración 2.14

- d. **Filtros pasa Alto:** Enfatizan las altas frecuencias, para mejorar o afiliar las características lineales como carreteras, fallas, o límites en general. Realizan por tanto el efecto contrario a los filtros pasa-bajos, eliminando estas las bajas frecuencias.

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & -8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Fig. 2.16: Ejemplo filtro paso alto: kernel

Otra forma de obtener un imagen así filtrada es sustraer a la imagen original, la misma imagen filtrada paso-bajos. Es lógico ya que si a la imagen le restamos los componentes de baja frecuencia, nos quedaremos con las de alta frecuencia.



Fig. 2.17 a) Imagen original; b) Imagen filtrada paso alto con el filtro de la ilustración 2.16.

- e. **Filtrado en Frecuencia:** En el dominio frecuencial también puede realizarse el proceso de filtrado, con mayor grado de comprensión de lo que se está viendo, ya que en una imagen en el dominio frecuencial se sabe dónde se encuentran los distintos rangos de frecuencias. De esta forma, en vez de realizar la convolucion, se efectúa su operación correspondiente en el dominio frecuencial con el producto

$$G(u, v) = H(u, v) \cdot F(u, v)$$

Los resultados que se obtienen son muy parecidos a los que se obtienen con el filtrado especial (convolucion) pero este caso se trabaja con otras variables y conceptos diferentes.

### 2.5.5.2 Filtros Morfológicos

La morfología matemática es un método no lineal de procesar imágenes digitales basándose en la forma. Su principal objetivo es la cuantificación de estructuras geométricas. Aquí los filtros también vienen definidos por su Kernel, pero no es un Kernel de convolucion sino un elemento estructurante

- a. **Dilatación:** es comúnmente conocido como “relleno”, “expansión”, o “crecimiento”. Puede ser usado para rellenar “huecos” de tamaño igual o menor que el elemento estructurante con la que se opera la dilatación.

Usado con imágenes binarias, donde cada pixel es 1 o 0, la dilatación es similar a la convolución. Sobre cada pixel de imagen se superpone el origen del elemento estructurante. Si el pixel de la imagen no es cero, cada pixel de la imagen no es cero, cada pixel que cae en la estructura es añadido al resultado aplicando el operador or.

Con la notación:

$$A \oplus B$$

Representando la dilatación de una imagen A por un elemento estructurante B, se puede escribir:

$$C = A \oplus B = U(A)$$

Donde  $(A)_b$  representa la traslación de A por b. intuitivamente, para cada elemento no cero  $b_i, j$  de B, A es trasladado  $i, j$  y sumado a C usando el operador  $\oplus$ . Por ejemplo:

0100		0110
0100		0110
0110	$\oplus$ 11	0111
1000		1100
0000		0000

Fig. 2.18: Ejemplo de dilatación con elemento estructurante en el (0,0).

Usado con imagen e escala de grises, la dilatación se efectúa tomando el máximo de una serie de sumas. Puede ser usado para implementar el operador de “máxima vecindad” con la forma de la vecindad dada en el elemento estructurante.



Fig. 2.19: a) Imagen original en escala de grises; b) imagen dilatada.

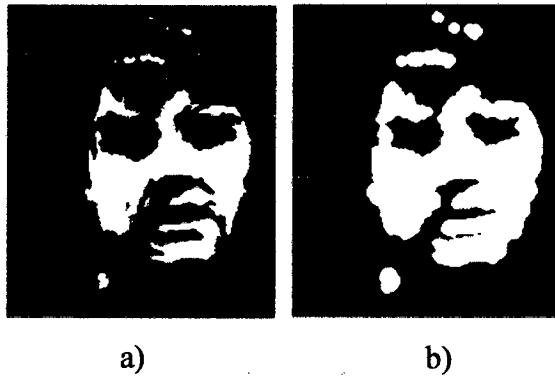


Fig. 2.20: a) Imagen binaria original; b) Imagen dilatada.

- b. Erosión:** La erosión es lo opuesto a la dilatación; realiza con el fondo lo que la dilatación al primer plano. También en este caso, existe un elemento estructurante que se utiliza para operar con la imagen. Los efectos son de “encogimiento”, “contracción” o “reducción”. Puede ser utilizado para eliminar islas menores en tamaño que el elemento estructurante.

Sobre cada pixel de la imagen se superpone el origen del elemento estructurante. Si cada elemento no cero de dicho elemento está conteniendo en la imagen, entonces el pixel de salida es puesto a 1. Haciendo  $A \ominus B$  como representación de la erosión de una imagen A por el elemento estructurante B, como representación de la erosión de una imagen A por el elemento estructurante B, se puede definir.

$$C = A \ominus B$$

Donde  $(A) \ominus b$  representa la traslación de A por b. B. puede ser visto como una “sonda” que se desliza a lo largo de toda la imagen A, testando la naturaleza espacial de A en cada punto. Si B trasladado i, j puede ser contenido en A (poniendo el origen de B en i, j), entonces la erosión de A por B. por ejemplo:

0100		0000
0100		0000
1110	=11	1100
1000		0000
0000		0000

Fig. 2.21: Ejemplo de erosión con elemento estructurante en (0,0).

Usando en imágenes en escala de grises, la erosión se efectúa tomando el mínimo de una serie de diferencias. Puede ser usado para implementar el operador de “mínima vecindad” con la forma de la “vecindad” dada por el elemento estructurante.

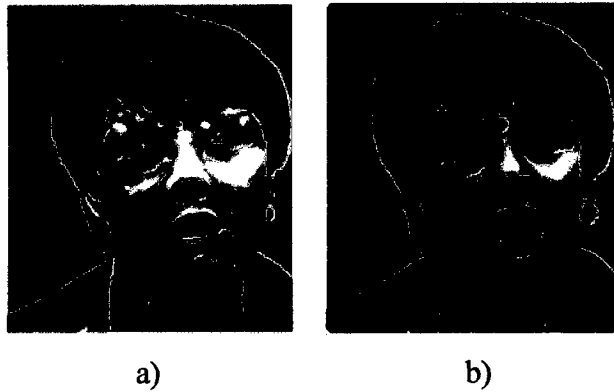


Fig. 2.22: a) Imagen original en escala de grises; b) imagen erosionada

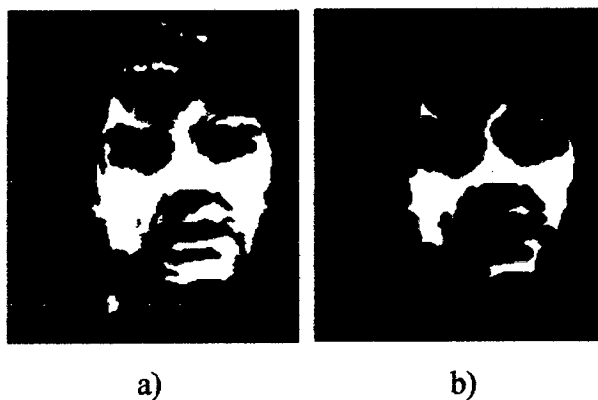


Fig. 2.23: a) Imagen binaria original; b) Imagen erosionada

- c. **Apertura y cierre:** La “apertura” (opening) de una imagen B por un elemento estructurante K, se define como:

$$(B * K) + K$$



a)

b)

Fig. 2.24: a) imagen binaria original; b) Imagen tras la apertura.

El “cierre” (closing) de la imagen B por el elemento estructurante K se define como:

$$(B+k) * k$$



a)

b)

Fig. 2.25: a) imagen binaria original; b) Imagen tras el cierre.

El resultado de aplicar interactivamente dilataciones y erosiones es la eliminación del detalle específico en la imagen menor que el elemento estructurante, si la distorsión geométrica global de características no suprimidas. Por ejemplo, “abrir” una imagen con una estructura en disco, suaviza los contornos, rompe istmos y elimina pequeñas islas, picos cabos. “cerrar” una imagen con un elemento estructurante en forma de disco, elimina pequeños agujeros y rellena brechas en los contornos.



### **2.5.5.3 Filtros de Textura**

Muchas imágenes contienen regiones caracterizadas por variaciones del nivel de gris, más que por un valor único de grises. La “textura” se refiere precisamente a la variación espacial de nivel de gris de una imagen como función de escala espacial. Para que los píxeles de una determinada área puedan ser definidos como texturalmente diferentes, sus niveles de gris deben ser más homogéneos como unidad que áreas de diferente textura.

De todas formas, el concepto de textura es bastante intuitivo. Es tan difícil definirlo como calcularlo. Mientras que el nivel de gris de un píxel está perfectamente definido y localizado, la textura es más elusiva. La definición que se ha dado es buena, pero a la vez vaga. De todas formas no hay ninguna mejor, y los filtros que se describen a continuación sirven de aproximación para trabajar con esta característica.

### **2.5.5.4 Filtro de Recorrido**

También llamados “de rango”. Este filtro sustituye el valor central de la ventana de procesamiento por la diferencia entre el valor máximo y mínimo (el recorrido, estadísticamente hablando) de los píxeles contenidos en esa ventana.

El recorrido será un valor pequeño para zonas “planas” o textualmente uniformes, y será alto en zonas de alta variabilidad. El tamaño de la ventana debe ser suficientemente grande como para incluir un número suficiente de puntos, según la escala a la que queramos trabajar, esto significa que debe ser mayor que el tamaño de cualquier pequeño detalle que pueda estar presente. El resultado es una imagen donde el valor de cada punto representa la textura, y diferentes regiones pueden ser distinguidas por diferentes niveles de gris.

### **2.5.5.5 Filtro RMS (Root-Mean-Square)**

Este filtro de textura calcula primero la varianza de los valores de la ventana y sustituye el valor central por el RMS de los píxeles de la ventana de proceso.

### **2.5.5.6 Operadores del Momento**

El primer y segundo momento son simples medidas de textura, utilizando los “momentos” del histograma de la ventana de proceso. El primer momento es una medida del contraste de la ventana. El segundo momento es una medida de la homogeneidad de la misma. Las imágenes resultantes pueden ser escaladas para crear una imagen que discrimina entre varias texturas.

## **2.6 DETECCIÓN DE CARAS**

A continuación se va a realizar una breve descripción de los métodos que pueden utilizarse para la detección de cara en una imagen, de manera que una vez detectada la cara se puedan aplicar los algoritmos de reconocimiento que se van a ver posteriormente.

Como se comentó anteriormente el problema del reconocimiento de caras humanas se pueden dividir en dos subtareas que deben resolverse por separado.

La primera fase es el proceso de detección de caras, es decir, dada una imagen determinada si en ella aparecen subimágenes que representan caras humanas y localizarlas para su posterior tratamiento. Tras esta fase viene la de reconocimiento de caras humanas, donde se trata de asignar una identidad a las caras obtenidas en la fase anterior. Estas dos fases deben ir siempre unidas, no es posible un reconocimiento de caras si estas no han sido previamente detectadas.

### **2.6.1 Técnicas basadas en Rasgos**

Estas técnicas explotan propiedades aparentes de la cara tal como color de la piel y la geometría facial. La detección de la cara se resuelve manipulando medidas de distancia, ángulos de rasgos visuales en la imagen. Lo más importante en este tipo de técnicas es decidir que rasgos de la cara interesan para su estudio.

El problema de detectar caras en una imagen tiene diferentes aspectos problemáticos. Algunas de ellas son las siguientes:

- Pose y orientación de la cara.
- Presencia de la cara.
- Expresión de la cara.
- Problemas de iluminación (no uniformidad).
- Condiciones generales de la imagen (ruido, fondo complejo).
- Cantidad desconocida de caras en la imagen.

### **2.6.2 Análisis de bajo Nivel**

Son técnicas que trabajan a nivel de píxel. Hay diversas técnicas dentro de este apartado, las más características son:

**a. Detección de bordes:** La idea se basa en analizar las líneas que componen los bordes de una cara y utilizarlas para detectar los rasgos faciales. El algoritmo sigue los siguientes pasos:

- Detectar los bordes de la imagen.

Una vez obtenidos los bordes, se procede a efectuar un adelgazamiento a fin de obtener para cada borde una línea de un píxel de ancho que lo represente.

- Filtrado de componentes.

El algoritmo se queda ahora solo con las componentes que sean más susceptibles de formar parte de una cara. Por ejemplo, buscando líneas que en conjunto se asemejen a una elipse de determinadas proporciones de ancho y alto.

- Etiquetado.

Una vez obtenidas dichas componentes, se etiqueten como lado derecho de la cara, lado izquierdo, línea del pelo, etc.

Las componentes etiquetadas se combinan para formar posibles candidatos para ser una cara, decisión que toma una función de coste, que utiliza la proporción áurea para sus cálculos. Este algoritmo es ineficiente si la cara no está de frente.

- b. Información de grises:** Trabaja sobre la idea de que rasgos faciales, tales como las cejas, las pupilas y los labios aparecen como zonas más oscuras de la imagen que las compone de las siguientes partes:

- Aumentar el contraste de la imagen.

De esta forma se resalta aún más la diferencia de luminosidad entre las citadas partes de la cara.

- Umbralización (Thresholding).

El algoritmo se queda solo con las zonas de la imagen cuyo valor gris supere un cierto umbral.

- Detección de caras mediante el uso de plantillas ponderadas.

Hasta aquí se ha obtenido una imagen compuesta por multitud de “manchas” negras. Este paso trata de comparar la distribución de esas manchas con las manchas “tipo” de una cara, usando plantillas. Hay varias propuestas de plantillas, algunas se basan en detectar primero las zonas de los ojos y a partir de ahí intentar detectar el resto de componentes. Otros métodos tratan de buscar máximos locales, como la punta de la nariz.

Este algoritmo produce diferentes resultados en función del color de la piel del sujeto que aparece en la imagen. Para corregir este problema, existe una técnica similar aunque más potente que usa la información del color.

En este proyecto se van a utilizar técnicas de este tipo para la detección de la cara. Como se ha dicho, las técnicas que usan la información del color son más potentes, así que aprovechando que las imágenes de las que se dispone son en color, se utilizara esta información.

Posteriormente en el capítulo de Diseño e Implementación se explica con más detalle las técnicas concretas utilizadas para la detección, tanto de zonas de piel como de zonas como los ojos y la boca, y su aplicación.

- c. **Video:** En una secuencia de video es más factible la localización de objetos en la imagen. Una de las mejores formas es mediante diferencia de fotogramas. Existen técnicas que miden variaciones verticales y horizontales para encontrar los ojos. Detectar contornos de una escena en movimiento es más sencillo. Se utilizan filtros espacios temporales de gaussiana para encontrar los bordes de la cara y el cuerpo.
- d. **Medidas Generalizadas:** se basan en el hecho de que la forma de la cara es simétrica y se utilizan medidas de simetría de la imagen.

### 2.6.3 Análisis de Rasgos

El problema del análisis a bajo nivel es que puede proporcionar información ambigua, por ejemplo, si aparecen en la imagen objetos que tengan un color similar al del modelo de color de piel utilizado.

$$b = \frac{NV_{max}}{ND_{max} - ND_{mtn}} +$$

El análisis de rasgos se basa en la geometría de la cara para caracterizar y posteriormente verificar rasgos a fin de evitar dicha ambigüedad. La búsqueda de rasgos plantea los posibles algoritmos:

- Búsqueda de la parte superior de la cabeza.

Se efectúa una hipótesis sobre lo que puede ser una posible línea del pelo en lo alto de la frente. Puede ser difícil si la persona tiene pelo cubriendo zonas de la frente.

Búsqueda de los ojos. A partir de dicha línea efectúa un barrido hacia abajo tratando de buscar zonas donde la densidad de gris aumente y disminuya bruscamente en el plano horizontal. Dichas zonas corresponden con las pupilas. Falla si el individuo usa gafas, si uno de los ojos no aparece por cualquier motivo, o si la imagen está rotada.

Uso de plantillas flexibles. La distancia entre la línea del pelo y el plano de los ojos se usa como medidas de referencia para inicializar una plantilla flexible que cubre el resto de rasgos, como la nariz y la boca. La plantilla trata entonces de ajustarse a dichos rasgos usando una función de costes basada en bordes.

- **Análisis de Constelaciones**

Se basa en el uso de modelo probabilística que estudia la posición espacial de los rasgos faciales, intentando buscar patrones que se asemejan a una cara.

## **2.6.4 Técnicas basadas en la Imagen**

En estas técnicas el objeto de estudio es la imagen misma. El conocimiento previo se incorpora implícitamente en esquemas de entrenamiento. Se trabaja directamente con una representación de la imagen a la que se le aplican algoritmos de gradientes de una función de energía.

### **2.6.4.1 Métodos basados en subespecies**

Consideran las imágenes de caras humanas como un subespecie lineal de un espacio mayor (de todas las imágenes). La base del método es la siguiente:

Construir la base canónica. Partiendo de un conjunto de imágenes que representan caras (y solo caras), se encuentran los componentes principales de una cara, expresados en términos de auto vectores, aquí llamados eigenfaces.

Cada cara del conjunto anterior puede ser aproximadamente por una combinación lineal de las eigenfaces usando los pesos apropiados.

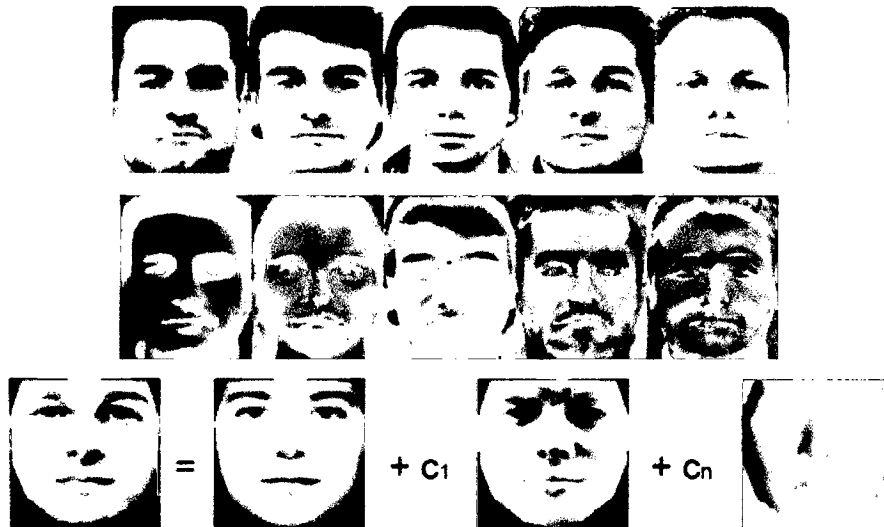


Fig. 2.26: a) Imágenes de caras; b) Eigenfaces; c) Aproximación de una cara mediante combinación lineal de eigenfaces.

**Detección de caras.** Tomando la imagen global como de partida, se intentan representar todos los elementos de la imagen en el espacio e las eigenfaces. Cada fracción e la imagen representada produce un error residual llamado “distancia de espacio de las caras” (DFFS: Distance from faces space). Observando los valores mínimos de estas distancias se puede determinar la posición de las caras.

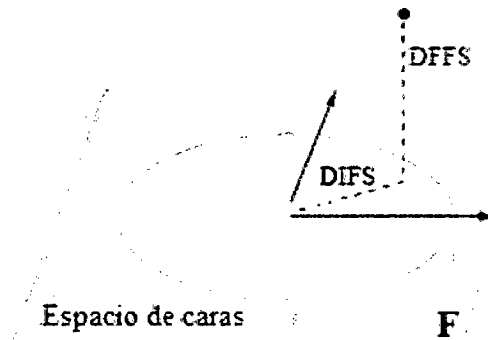


Fig. 2.27: Espacio de caras: DFFS (Distance from faces space), DIFS (Distance within face space).

## 2.7. RECONOCIMIENTO DE CARAS

En este apartado se va a realizar un breve estudio de diferentes algoritmos utilizados para el reconocimiento de caras. Los algoritmos que se van a ver se pueden clasificar dentro del grupo:

### 2.7.1. Métodos Holísticos

Utilizan toda la imagen de la cara como entrada al sistema de reconocimiento, siendo ésta la unidad básica de procesamiento.

Métodos basados en características locales: Se extraen características locales, como ojos, nariz, etc. Sus posiciones y estadísticas locales constituyen la entrada al sistema de reconocimiento.

También existen métodos híbridos que combinan técnicas holísticas y locales.

Para poder aplicar y entender muchos de los métodos que se van a ver a continuación es necesario tener unos conocimientos previos sobre álgebra y estadística que se van a dar por sabidos.

### **2.7.1.1. Análisis de Componentes Principales (Principal Component Analysis PCA)**

El análisis de componentes principales es un método típico en el análisis de datos multivariantes cuyo objetivo es la reducción de la dimensionalidad de los mismos.

Lo que trata de hacer es analizar si dadas  $c$  muestras de un conjunto de  $n$  valores se puede representar la información por un número menor de variables, construidas como combinaciones lineales de las originales. El funcionamiento de este método puede resumirse en el siguiente algoritmo:

- Obtener un conjunto de datos de dimensión  $n$ .
- Calcular la media de los datos y restársela a cada uno de ellos, de esta forma se tiene unos datos cuya media es cero.
- Calcular la matriz de covarianza.
- Calcular los eigenvectores (vectores propios) y eigenvalores (valores propios) de la matriz de covarianza.
- Elegir las componentes y formar un vector característico. Se ordenan los eigenvectores de mayor a menor, y se eligen los  $p$  eigenvectores ( $p < n$ ) correspondientes a los mayores eigenvalores. Así se tiene un espacio de menor dimensión.
- Obtener el nuevo conjunto de datos. Los datos originales se multiplican por el vector característico, así se tendrán los datos en términos de los eigenvectores elegidos.

PCA es un método general de análisis de datos y se aplica en el reconocimiento de caras con alguna variación en el método, llamándolo eigenfaces. Por tanto el método que realmente se va a describir en este aparato es el de eigenfaces.

Una imagen puede considerarse como un vector de píxeles donde el valor de cada componente es un valor en escala de grises, por ejemplo una imagen de  $256 \times 256$  ( $N \times N$ ) será un vector de dimensión  $65536$  ( $N^2$ ), es decir la imagen estará en un espacio de dimensión  $65536$ . Las imágenes de caras no están distribuidas aleatoriamente en este espacio y pueden ser descritas como un subespacio de menor dimensión.

La idea del análisis de componentes principales es encontrar los vectores que mejor representan las imágenes de caras dentro del espacio completo de imágenes.



Estos vectores definen el subespacio de imágenes de caras (facespace).

Cada vector de longitud  $N^2$  es una combinación lineal de las imágenes originales.

Estos vectores son los eigenvectores de la matriz de covarianza del espacio original de imágenes de caras y se llaman eigenfaces porque son parecidos a una cara.

Si se tiene un conjunto de imágenes de caras  $X_1, X_2, \dots, X_m$ , se puede formar la matriz  $X$  poniendo como vectores columna cada imagen (tendremos una matriz de dimensión  $N^2 \times m$ ). Se calcula la media del conjunto de imágenes como:

$$\mu = \frac{1}{m} \sum_{i=1}^m X_i$$

Se obtiene la diferencia entre cada vector y la media:  $\phi_i = X_i - \mu$ . Aplicando PCA se busca un conjunto de  $m$  vectores ortonormales  $u_k$  que describen la distribución de los datos.

Los vectores  $u_k$  son los eigenvectores y los valores:

$$h_k = \frac{1}{m} \sum_{i=1}^m (u_k^T * \phi_i)$$

Son los eigenvectores de la matriz de covarianza:

$$S = \frac{1}{m} \sum_{i=1}^m \phi_i * \phi_i^T = A * A^T$$

Donde  $A = [\phi_1, \phi_2, \dots, \phi_i]$

El cálculo computacional que esto implica es muy elevado, por eso si  $m < N^2$  solo habrá  $m - 1$  eigenvectores, facilitando así los cálculos y pudiendo resolver el problema utilizando combinaciones lineales de las imágenes. Se calcula la matriz  $L = A * A^T$  de dimensión  $(m * m)$  y se buscan los  $m$  eigenvectores  $V_1$ . Estos vectores determinan la combinación lineal de las  $m$  imágenes del conjunto de entrenamiento:

$$u_k = \sum_{i=1}^m v_{li} * \phi_i$$

Para el reconocimiento de una cara se debe seguir el siguiente procedimiento:

1. Obtener un conjunto inicial de imágenes de caras.



Fig. 2.28: Conjunto inicial de caras.

2. Calcular la matriz L. encontrar sus eigenvectores y eigenvalores, y elegir los M (en la práctica se puede utilizar  $M < m$ ) eigenvectores que tienen mayores eigenvalores.
3. Combinar las imágenes del conjunto de entrenamiento para obtener los eigenfaces.

$$u_k = \sum_{i=1}^m v_{li} * \phi_i$$

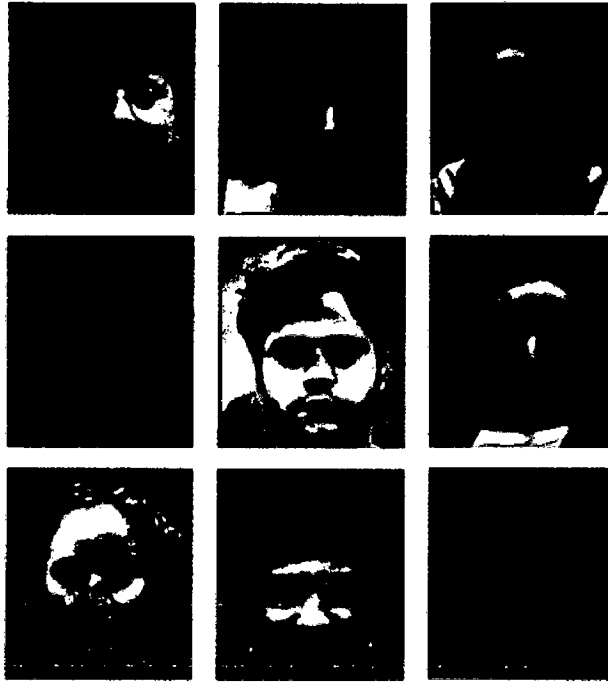


Fig. 2.29: Eigenfaces calculados son las imágenes iniciales ( $M = 7$ )

4. Para cada cara conocida calcular el vector clase  $\Omega^T = [\omega_1, \omega_2, \dots, \omega_m]$  donde

$$\omega_1 = u_k^T(x - \phi)$$

5. Para una nueva cara calcular su vector  $\Omega$  y calcular la distancia a cada clase:

$$\varepsilon_k^2 = \|(\Omega - \Omega_k)\|$$

Una cara es clasificada como perteneciente a la clase  $k$  cuando el mínimo  $\varepsilon_k$  está por debajo de un umbral. En cualquier otro caso la cara es clasificada como desconocida.

Si  $\min(\varepsilon_k) < 10000000 \rightarrow \text{cara} \in \text{clase } k$



$\min(\varepsilon_k) = 6985799, k=1$



$\min(\varepsilon_k) = 57188080, \text{ no clasificada}$

Fig. 2.30: Ejemplo resultado clasificación.

En este proyecto se va a utilizar PCA como método de reducción de dimensionalidad. No se utilizara el algoritmo completo para el reconocimiento de caras, sino que se aplicará para extraer características de cada imagen, y estas características pasaran a una etapa de clasificación. En el capítulo de Diseño e Implementación se explicará y se aplicará esta extracción de características mediante PCA.

### **2.7.1.2. Análisis de Componentes Independientes ICA**

ICA (Independent Component Analysis) es una generalización del método PCA. Este método trata de descomponer una señal observada en una combinación lineal de fuentes independientes. Mientras que PCA de correlaciona las señales de entrada utilizando estadísticos de segundo orden, ICA minimiza mayores ordenes de dependencia.

Se tiene una matriz de variables independientes (fuentes):  $S = (S_1, \dots, S_n)$  y una matriz de observación  $X$ . En esta matriz de observaciones, cada columna es el resultado de un experimento aleatorio, y en cada fila se tiene el valor de una prueba de ese experimento. Como se ha dicho el método ICA trata de descomponer la señal observada en una combinación de fuentes independientes, la matriz de combinación (desconocida) se llamará  $A$ :  $X = A \cdot S$

Con el algoritmo ICA se busca la matriz de separación  $w$ , que cumple:

$$U = W \cdot X = W \cdot A \cdot S,$$

Donde  $U$  es la estimación de máxima probabilidad (ML) de las componentes independientes. Hay formas de implementar el método ICA para el reconocimiento de caras:

Se puede poner en cada fila de la matriz  $X$  una imagen diferente, así se tendrá que cada imagen es una variable aleatoria y los píxeles son pruebas.

Otra opción es transponer la matriz  $X$  y tener en cada columna una imagen, de manera que es este caso, los píxeles son variables aleatorios y cada imagen una prueba.

Se va a ver como se trabajan si se decide optar por la primera opción, es decir, en cada fila la matriz X hay una imagen, y en las columnas están los píxeles. Al aplicar ICA se encuentra una matriz W tal que las columnas de U son tan estadísticamente independientes como es posible. Las imágenes fuentes estimadas por las columnas de U son utilizadas como imágenes base para la representación de caras.

Lo que se suele hacer para reducir la complejidad computacional es aplicar ICA sobre un conjunto de m (con  $m < n^\circ$  filas) combinaciones lineales de las imágenes (aplicando por ejemplo PCA), obteniendo una matriz de m imágenes fuente independiente en las columnas de U. Para determinar los coeficientes de la combinación de las imágenes base se siguen los siguientes pasos:

$P_m$ : Matriz que contiene los m componentes principales.

$R_m = X \cdot P_m^T$  Representación de X en la base  $P_m$ .

$X = R_m \cdot P_m^T$  Aproximación de mínimo error cuadrático.

$$W_2 = 2 \cdot (\cos(X))^{(-\frac{1}{2})} W_1 = W \cdot W_2 W_{1 \cdot P_m^T} = U \quad X = R_m \cdot W_1^{-1} \cdot U$$

De esta manera se obtienen los coeficientes como  $B = R_m \cdot W_1^{-1}$ . Se puede reconocer una cara evaluando los coeficientes B, utilizando para ello el algoritmo de vecinos más cercanos (nearest nighbor) con el coseno como medida de similitud:

$$C = \frac{b_{test} \cdot b_{train}}{\|b_{test}\| \cdot \|b_{train}\|}$$

### 2.7.1.3. ANALISIS DISCRIMANTE LINEAL: LDA

Este método es una técnica de aprendizaje supervisado para clasificar datos. La idea central de LDA es obtener una proyección de los datos en un espacio de menor (o incluso igual) dimensión que los datos entrantes, con el fin de que la separación de las clases sea la

mayor posible. Es una técnica supervisada ya que para poder buscar esa proyección se debe entrenar el sistema con patrones etiquetados.

El escenario de trabajo necesario para aplicar este método al reconocimiento de caras se basa en que se dispone de un conjunto de caras de entrenamiento ( $X_i$ ) compuesto por un grupo de personas con distintas expresiones faciales y con diferentes vistas. Por definición todas las instancias de caras de la misma persona están en una clase (de tamaño ( $N_c$ )), y las caras de otras personas diferentes pertenecen a distintas clases (habrá  $c$  clases), teniendo así el espacio de entrenamiento separado en grupos.

Además todas las instancias en el conjunto de entrenamiento deben estar etiquetadas.

Se calcula la matriz de dispersión dentro de una clase ( $S_b$ ):

$$S_b = \sum_c N_c (\mu_c - \mu) \cdot (\mu_c - \mu)^T$$

$$S_b = \sum_c \sum_{ie} (x_i - \mu_c) \cdot (x_i - \mu_c)^T$$

Siendo  $\mu_c$  la media de cada clase y  $\mu$  la media total.

Se trata de obtener un vector un vector de proyección  $w$ , que haga la razón entre la dispersión intra-clase y la dispersión inter-clase sea máxima. Habría que maximizar la siguiente función objetivo:

$$J(W) = \frac{W^T \cdot S_B \cdot W}{W^T \cdot S_w \cdot W}$$

El vector  $w$  que maximiza esta función será aquel que cumpla la siguiente ecuación:

$$S_B \cdot W = \lambda \cdot S_w \cdot W$$

Si la matriz  $S_w$  no es singular (tiene inversa) se tiene un problema de valores propios para la matriz  $S_B^{-1} \cdot S_B$ , que puede sustituirse en la función objetivo:

$$S_w^{-1} \cdot S_B = \lambda \cdot W \rightarrow J(W) = \frac{W^T \cdot S_B \cdot W}{W^T \cdot S_w \cdot W} = \lambda_K \frac{W_K^T \cdot S_B \cdot W_K}{W_K^T \cdot S_w \cdot W_K}$$

Con  $k = 1, \dots, d$  (siendo  $d$  la dimensión de las imágenes).

En la expresión anterior se puede ver que el vector propio que maximiza la función es aquel que tenga un mayor valor propio (mayor  $\lambda_k$ ).

#### **2.7.1.4. Métodos basados en Kernels**

Estos métodos son una generalización de los métodos de análisis de componentes (PCA, ICA, LDA).

En los métodos de componentes se construye un subespacio que cumpla determinadas restricciones y luego se elige una base que lo genere. En los métodos de Kernels se tienen en cuenta momentos de mayor orden sin tener un costo computacional excesivamente grande.

Se lleva el problema de clasificación a un espacio de mayor dimensión donde las clases sean linealmente separables. Para esto se realiza lo siguiente:

Se mapean los vectores de entrenamiento a través de una función no lineal que lleva los puntos a un espacio de mayor dimensión.

Se plantea un problema equivalente al problema de PCA, ICA o LDA en dicho espacio.

Se resuelve el problema equivalente, utilizando el Kernel trick, que es una manera simplificada de resolver el problema de PCA, ICA o LDA en el espacio de mayor

dimensión. Si cumplen determinadas condiciones particulares, se pueden realizar todos los cálculos de la resolución del problema equivalente sin necesidad de mapear los vectores en el espacio de mayor dimensión para esto existen diferentes funciones, llamadas núcleos (Kernels), que lo hacen posible.

#### 2.7.1.5. Evolutionary Pursuit: EP

Este método es un tipo de algoritmo genético que trata de encontrar una base de cara a través de la rotación de ejes definidos en un espacio blanco PCA adecuado. La evolución es conducida por una función de fitness que depende de la precisión de la clasificación y de la habilidad para generalizar.

El algoritmo EP se utiliza para buscar entre las diferentes rotaciones y vectores base para encontrar un subconjunto de vectores óptimo (que tenga buena precisión en la clasificación y habilidad para generalizar).

La base óptima evoluciona desde un conjunto de vectores de una base  $a_1, a_2, \dots, \frac{a_m(m-1)}{2}$ .

La función de fitness guía al algoritmo sobre cómo elegir los elementos que formarán la siguiente generación. Si  $F = a_1, a_2, \dots, \frac{a_m(m-1)}{2}, a_1, a_2, \dots, a_m$ , representan los

parámetros que son desarrollados por el algoritmo, entonces la función de fitness es:  $\zeta(F) = \zeta_a(F) + \lambda(\zeta)$ . donde  $\zeta_a$  es el término de clasificación,  $\zeta_s$  es el término de generalización y  $\lambda$  es una constante positiva que determina la importancia del segundo término respecto al primero.

Dado un conjunto de ángulos de rotación, la base de vectores después de la transformación es . Si en el algoritmo se eligen  $l$  vectores,  $n_1, n_2, \dots, n_n$  de la nueva base, el nuevo conjunto de características es  $W = [n_1, n_2, \dots, n_n] \cdot V$  donde  $V$  es el conjunto blanco de características. Así se tienen  $W_1, W_2, \dots, W_N$  clase con  $N_1$  imágenes cada clase. El algoritmo que se sigue para encontrar las EP-faces es:

Reducir la dimensión de los datos mediante PCA.

Transformar el espacio anterior para que sea blanco.



Realizar el siguiente bucle hasta llegar a un número máximo de interacciones o encontrar la solución buscada.

Realizar varias rotaciones entre pares de vectores de una base del espacio y luego seleccionar un conjunto de ellos. Codificar cada rotación mediante una representación en palabras de bits (en los algoritmos genéticos la representación se hace en función de bits).

Calcular la función de fitness para medir la precisión y generalización.

Calcular los ángulos y vectores que maximizan la función. Se guardan como la mejor solución hasta el momento.

Se itera a un nuevo subconjunto de ángulos y por lo tanto de vectores rotados.

Con la base óptima hallada se realiza el reconocimiento de caras mediante alguna medida de similitud.

# **CAPITULO III**

# **SISTEMA, DISEÑO Y**

# **DESARROLLO**

**TESIS PARA OBTENER EL TITULO PROFESIONAL  
INGENIERO ELECTRONICO Y TELECOMUNICACIONES  
UNIVERSIDAD NACIONAL DE PIURA**

### **3.1 METODOLOGIA DEL PROYECTO**

La metodología de trabajo se divide en los siguientes pasos:

#### **3.1.1 Aplicación de los Conceptos de Reconocimiento Facial**

Este es el primer paso, el cual consiste en conceptualizar los conceptos de sistemas de reconocimiento facial, en particular; los utilizados para el marcado de puntos característicos de la cara, de tal manera que nos permita tener una orientación clara al momento de desarrollar el algoritmo de reconocimiento.

#### **3.1.2 Desarrollo de la Detección y Marcado de Patrones Biométricos**

El desarrollo incluye los siguientes pasos:

- PRE procesamiento de las imágenes.
- Detección de los rasgos biométricos del rostro (ojos, nariz y boca)
- Extracción de dichos rasgos biométricos

#### **3.1.3 Desarrollo de la Identificación Biométrica.**

Esta parte consiste en utilizar las características extraídas de las imágenes generadas en la etapa anterior. El desarrollo incluye los siguientes pasos.

- Extracción de características (distancias euclídeas entre los puntos característicos).
- Registro del individuo en la Base de Datos
  - Patrones de Identificación Biométrica (XLS)
  - Archivo de imagen JPG
- Comparación de Similitud.
- Toma de decisiones.

### **3.2 ESQUEMA DEL SISTEMA**

#### **3.2.1 Sistemas biométricos.**

En la figura 3.2 se muestra el diagrama de bloques en forma general, del Sistema de Reconocimiento Biométrico. Se puede apreciar los tres módulos básicos en los cuales se ha centrado el desarrollo de la aplicación para el reconocimiento de rostros:

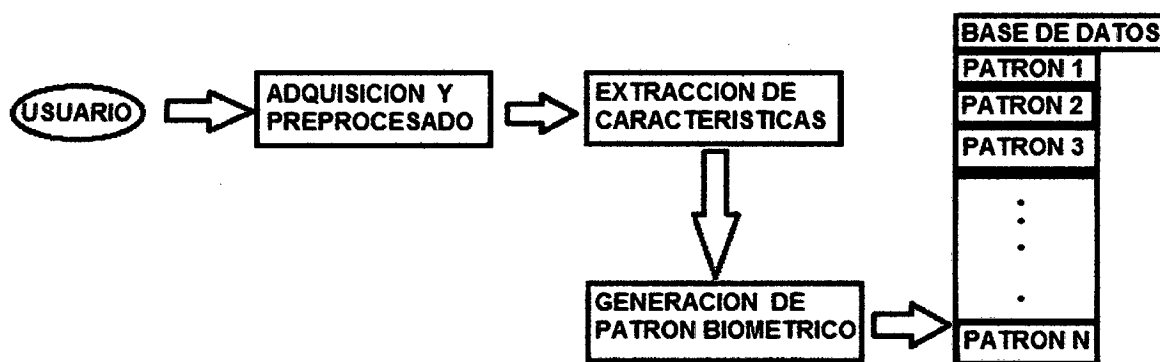
- La Base de Datos (Archivo XLS y JPG).

- El Módulo de Inscripción.
- El módulo de reconocimiento

El conjunto de estos módulos realiza las funciones necesarias para reconocer a un individuo, comparándolo con los patrones de identificación de los individuos almacenados en un archivo XLS relacionados con sus respectivas imágenes JPG.

- **Módulo de Inscripción**

Este módulo, el cual se muestra en la figura 3.1 está formado por el dispositivo de adquisición de las imágenes (WEBCAM) el cual a través de una imagen JPG proporciona la señal biométrica que caracteriza al individuo. Tras la adquisición de la señal biométrica se procede a la extracción de las características del rasgo biométrico del individuo (Método PCA usando Procesamiento de Imágenes con Matlab). Dichas características deberán expresar de una forma unívoca y compacta al individuo constituyendo su llamado patrón biométrico. El vector de características así formado, presenta menor dimensión que la señal previamente adquirida. De esta forma se realiza la codificación óptima de la señal en la que toda información irrelevante, que no contribuye al reconocimiento, es eliminada.



**Fig. 3.1 Modulo de inscripción.**

- **Base de datos - XLS/JPG**

El patrón biométrico extraído por el módulo de inscripción es almacenado en un archivo XLS, el cual será la base de datos del sistema de reconocimiento. La base de datos contendrá, por tanto, los patrones biométricos de los individuos registrados en el sistema.

- **Módulo de reconocimiento**

Este módulo se encarga de establecer la identidad del individuo que accede al sistema. Para ello, tras la adquisición del rasgo biométrico del individuo, se extraen las características y se obtiene el patrón biométrico, que, posteriormente, es comparado con los patrones almacenados en la base de datos. Los resultados de dichas comparaciones son cuantificados y valorados, permitiendo así la toma de decisiones respecto a la identidad del individuo en función de similitud obtenida.

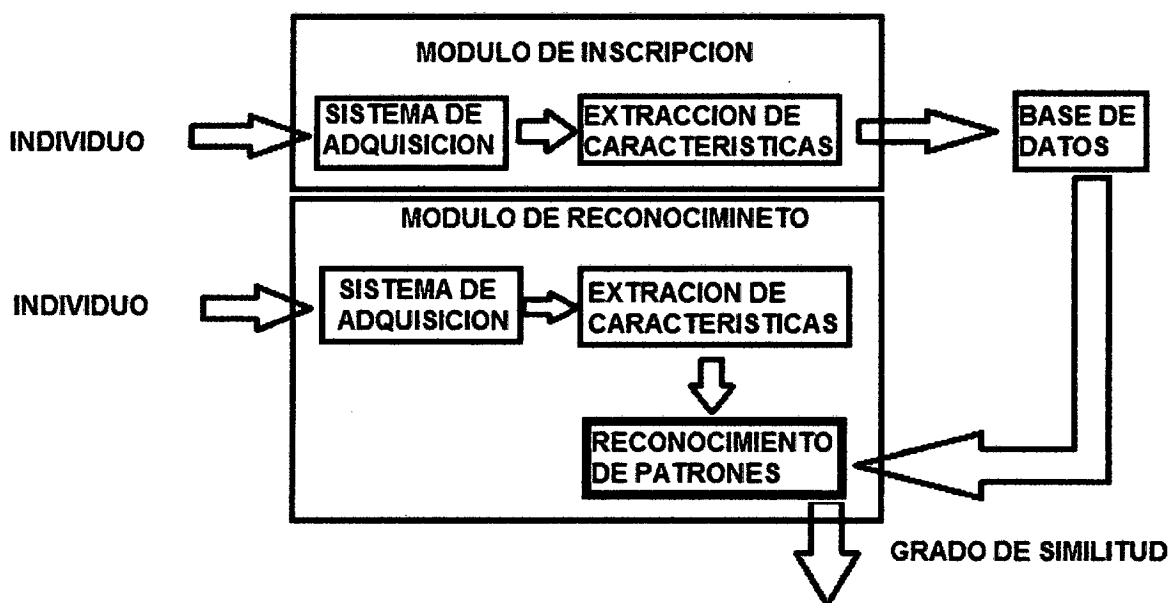


Fig. 3.2 Diagrama de bloques del Sistema Biométrico de Reconocimiento.

Desde el punto de vista del funcionamiento de los sistemas automáticos de reconocimiento de personas mediante rasgos biométricos, en cuanto a las perspectivas fundamentales del trabajo de identificación, este proyecto ha utilizado el sistema de reconocimiento en modo identificación.

### 3.2.2 MODO DE IDENTIFICACION

El objetivo es el de clasificar una realización determinada de un rasgo biométrico de identidad desconocida como perteneciente a uno de entre un conjunto de  $N$  posibles individuos. El esquema es el representado en la figura 3.3

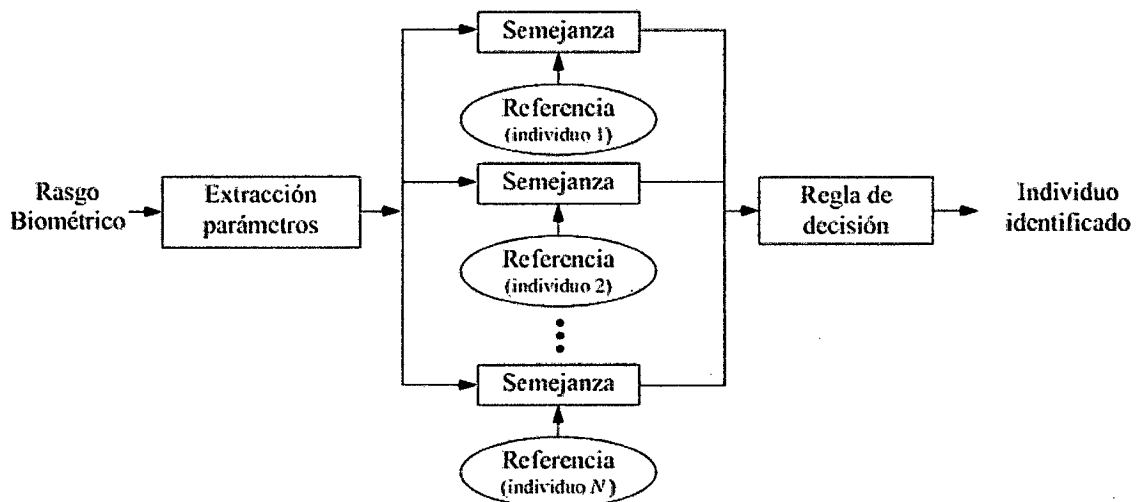


Fig. 3.3 Esquema modo de identificación.

Los errores que el sistema puede cometer durante el funcionamiento en modo de Identificación son:

- El sistema se decide por el identificador de un patrón erróneo de la base de datos. Por tanto, el usuario no aparece como primer candidato de la lista.
- El usuario que demanda ser identificado, a pesar de aparecer como primer candidato en la lista, no obtiene el grado de semejanza suficiente como para ser reconocido como tal, y en consecuencia, el sistema considera que no pertenece a la base de datos.

Dentro de estos sistemas, debemos diferenciar dos posibles casos:

- **Identificación en conjunto cerrado:** en este caso, el resultado del proceso es una asignación de identidad a uno de los individuos modelados por el sistema, y conocidos como usuarios. Existen, por tanto, N probables decisiones de salida posibles.
- **Identificación en conjunto abierto:** aquí debemos considerar una posibilidad adicional a las N del caso anterior: que el individuo que pretende ser identificado no pertenezca al grupo de usuarios, con lo que el sistema de identificación debería contemplar la posibilidad de no clasificar la realización de entrada como perteneciente a las N posibilidades.

### 3.2.3 DESCRIPCION DEL SISTEMA DESARROLLADO.

Para el cumplimiento de los objetivos del presente proyecto, se ha seguido el esquema que se muestra en la figura 3.4. En ella se puede apreciar dos bloques bien definidos con cada una de sus etapas antes de la detección de los parámetros los cuales se van a utilizar para la comparación del rostro en procesamiento y los almacenados en la base de datos. En el bloque superior de la figura 3.4 se encuentran las etapas de adquisición de la imagen a procesar y la que es inicialmente almacenada en la base de datos (archivo XLS) como el archivo input.jpg. Antes del preprocesado de esta imagen, se determinan las coordenadas de los parámetros a ser tomadas como referencia para la comparación de la imagen con las almacenadas en la base de datos como archivos imXXX.jpg (XXX es un número entero de 001 a 999 par el caso de investigación). Estos parámetros son referidos a las coordenadas Px y Py de la posición del centro del ojo derecho, el cual será el punto de comparación con la distancia al centro del ojo izquierdo el cual determinará el individuo en comparación. El bloque de preprocesado es aquel donde se determina si la imagen a comparar es un rostro humano, además que si es así, se procede a rotarlo de tal manera que el eje de los ojos quede horizontal de tal manera que la distancia de la coordenadas Py de los dos ojos del rostro en procesamiento sean iguales. El bloque inferior de la figura 3.4 esta referida a la detección y comparación de la imagen en procesamiento con las almacenadas en la base de datos. La detección de umbrales es aquella que determina las coordenadas Px y Py del ojo derecho del rostro y la etapa de detección y comparación de imágenes, establece ratios de comparación el cual determina el individuo que se encuentra en la base de datos.

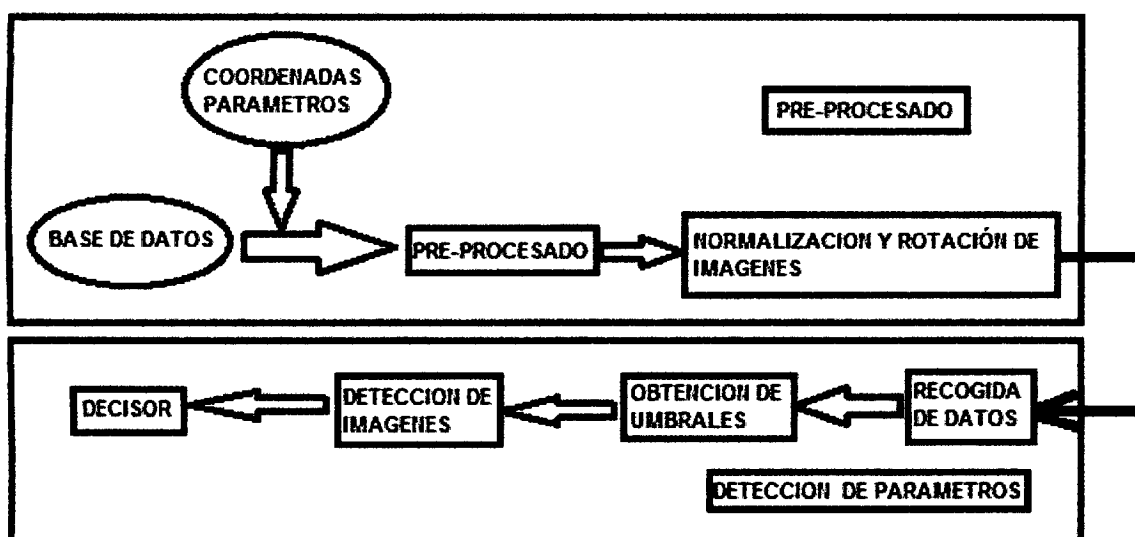


Figura 3.4 Esquema general para la detección de puntos característicos del rostro.

### 3.3 DISEÑO DEL PROGRAMA DE RECONOCIMIENTO E IDENTIFICACION

Para la identificación de un individuo utilizando procesamiento de imágenes, el presente proyecto de tesis plantea en su diseño dos Bloques:

- ✓ Diseño del Software PCA
- ✓ Diseño de la Interface HMI - PCA

#### 3.3.1 Diseño de Software PCA

- Ruta de la dirección de la DB o archivo XLS donde se almacenará los parámetros de comparación.
- Configuración y detección de la cámara de adquisición de imágenes o WEBCAM.
- Captura de la imagen y almacenamiento en la DB (carpeta de archivos imágenes) como archivo input.jpg.
- Procesamiento de la imagen input.jpg para determinar si corresponde a un rostro humano y determinación de las coordenadas Px y Py del eje del ojo derecho.
- Adicionar datos de la imagen de un rostro nuevo y sus parámetros en la DB (archivo XLX) así como la respectiva imagen (imXXX.jpg)
- Comparar la imagen en procesada con imágenes almacenadas en la DB
- Resultados PCA de la comparación de las imágenes.

#### 3.3.2 Diseño de Interface HMI

La interface desarrollada en la herramienta GUI del Matlab se muestra en la figura 3.5

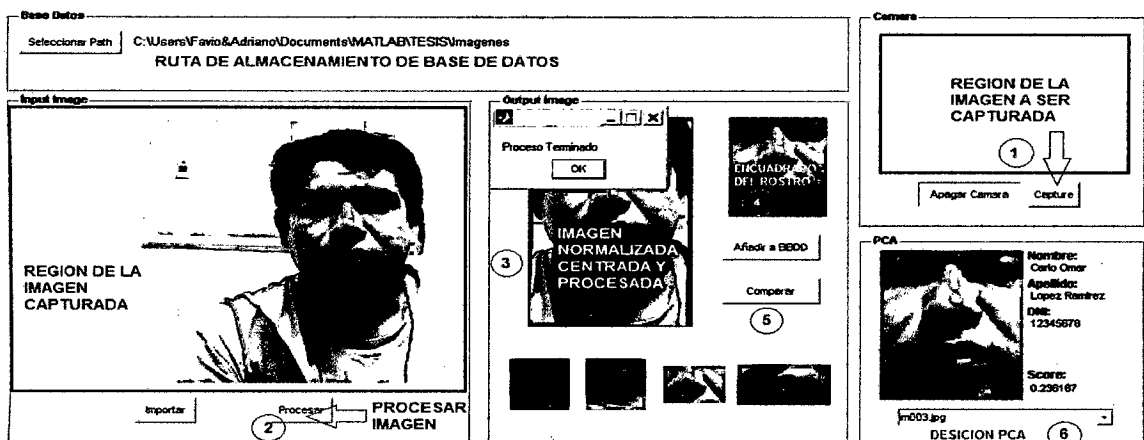


Fig. 3.5 Interface HMI - PCA Reconocimiento de Rostros



**Paso 1:** Área o región donde la imagen es capturada por la cámara webcam al momento de dar la orden.

**Paso 2:** Área o región donde la imagen es capturada, para luego ser procesada. En esta área también se importan imágenes de otras rutas que han sido almacenadas en otra base de datos.

**Paso 3:** Área o región donde la imagen es normalizada, centrada y procesada y en la cual se han obtenido los patrones de comparación.

**Paso 4:** Área o región donde se muestra el área de la imagen que contiene solo los ojos nariz y boca, además que la imagen es encuadrada.

**Paso 5:** Área o región donde se puede realizar las siguientes operaciones: Añadir a BBDD que corresponde a convertir el archivo procesado output.jpg a imXXX.jpg y almacenarla en la carpeta de imágenes de la base de datos. Y la otra operación es comparar la imagen output.jpg con cada una de las imXXX.jpg almacenadas en la DDBB.

**Paso 6:** Área o región donde se muestra la información de la imagen procesada respecto a la identidad del individuo y sus datos personales. La opción List muestra todas las imágenes de los individuos almacenados en la DDBB.

### 3.3.3 ALGORITMOS Y CODIGO DEL PROGRAMA PCA

#### 3.3.3.1 RUTA DE BASE DE DATOS

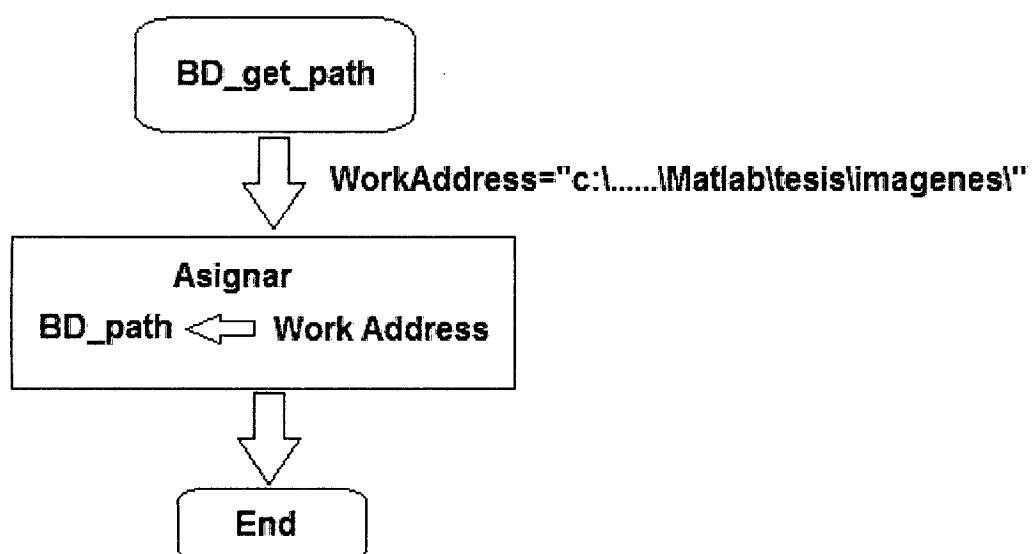


Fig. 3.6 Algoritmo de generación de Ruta de la DDBB.

Esta rutina se ejecutará siempre y cuando se presione el push\_botton "Seleccionar Path" que se muestra en la Fig. 3.5 y cuya interface como se muestra en la figura 3.6

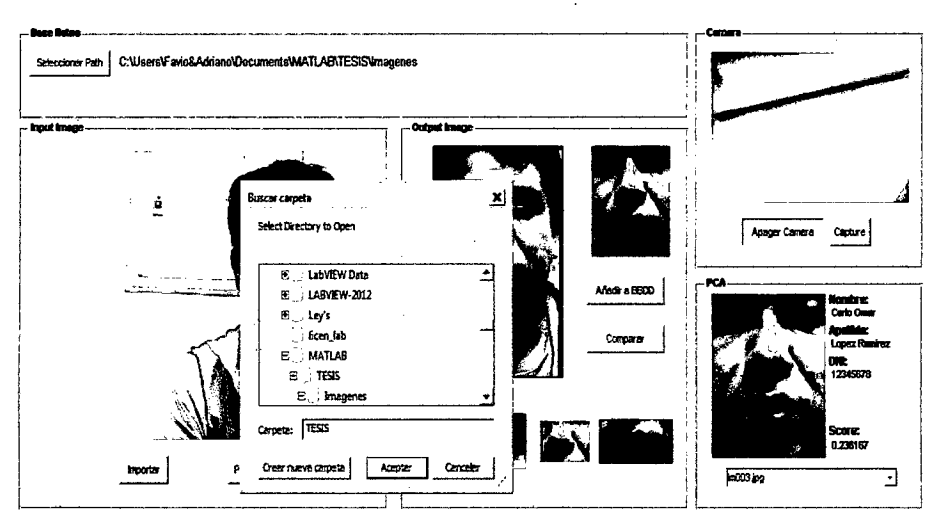


Fig. 3.6 Sub Menú para elegir ruta de ejecución del Programa PCA

### 3.3.3.2 CONFIGURACION Y DETECCION DE LA WEBCAM

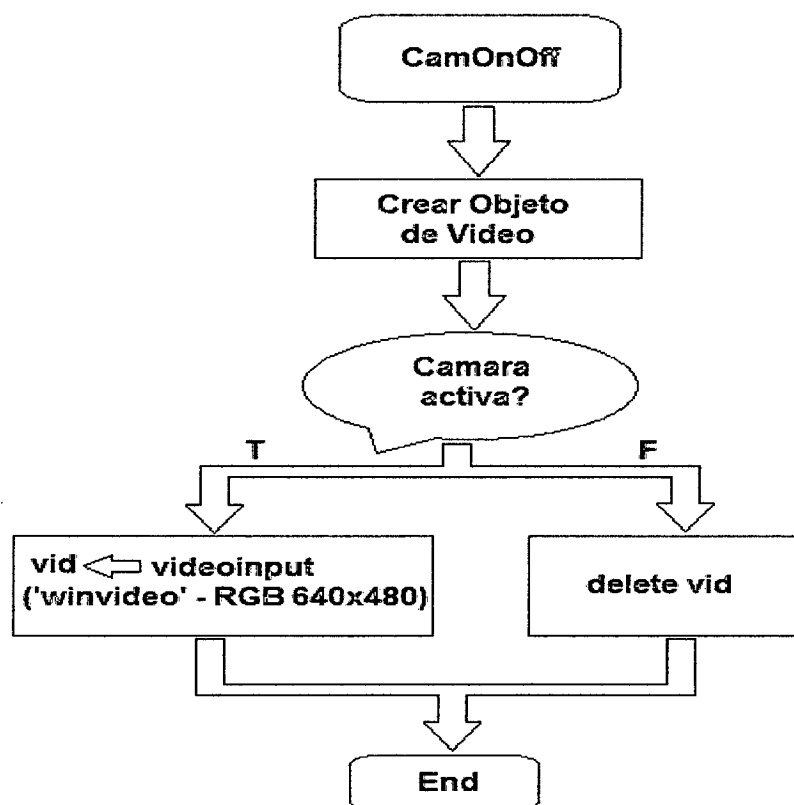


Fig. 3.7 Algoritmo de Activación/Desactivación de WEBCAM

En la zona de Cámara que se muestra en la figura 3.5 (recuadro superior derecho) hay un push\_button de Encender/Apagar Cámara. Cuando se activa el push\_button, existe una variable denominada vid a la cual se asigna los parámetros Adaptor Name, Device IDs y

formato. El Adaptor Name está referido al objeto de entrada de video el cual representa la conexión entre el Matlab y el dispositivo de adquisición de imágenes. El Adaptor Name para esta aplicación es una DLL de Windows denominada winvideo y DeviceIDs es "1" por que solo usamos una sola cámara. El formato para las imágenes JPG es RGB 640x480.

### 3.3.3.3 CAPTURA Y ALMACENAMIENTO DE LA IMAGEN

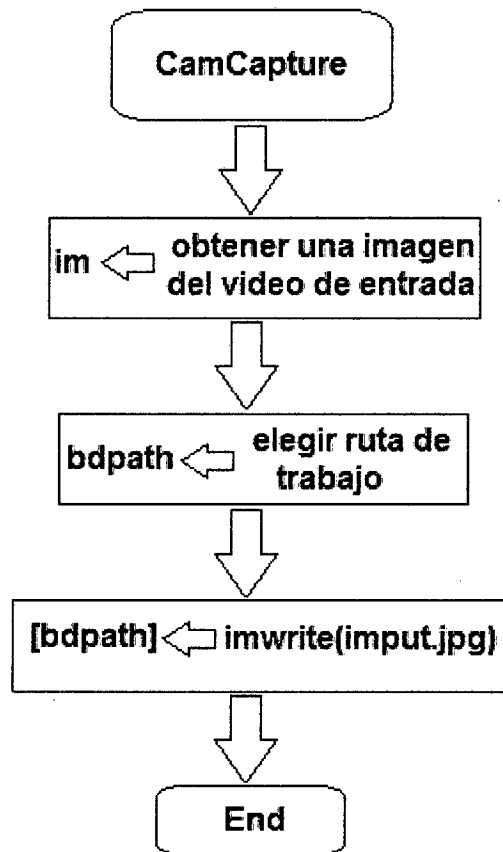


Fig. 3.8 Algoritmo de Captura de Imagen

El algoritmo de captura de imágenes, captura una imagen instantánea (snapshot) del adaptador de video al presionar el push\_botton "Capture" en la interface HMI - PCA que se muestra en la figura 3.5. Ésta imagen se visualiza en la zona "input imagen" del HMI y se almacena en la capeta Imágenes con el nombre de "input.jpg" para su posterior procesamiento.

### 3.3.3.4 PROCESAMIENTO DEL ARCHIVO INPUT.JPG

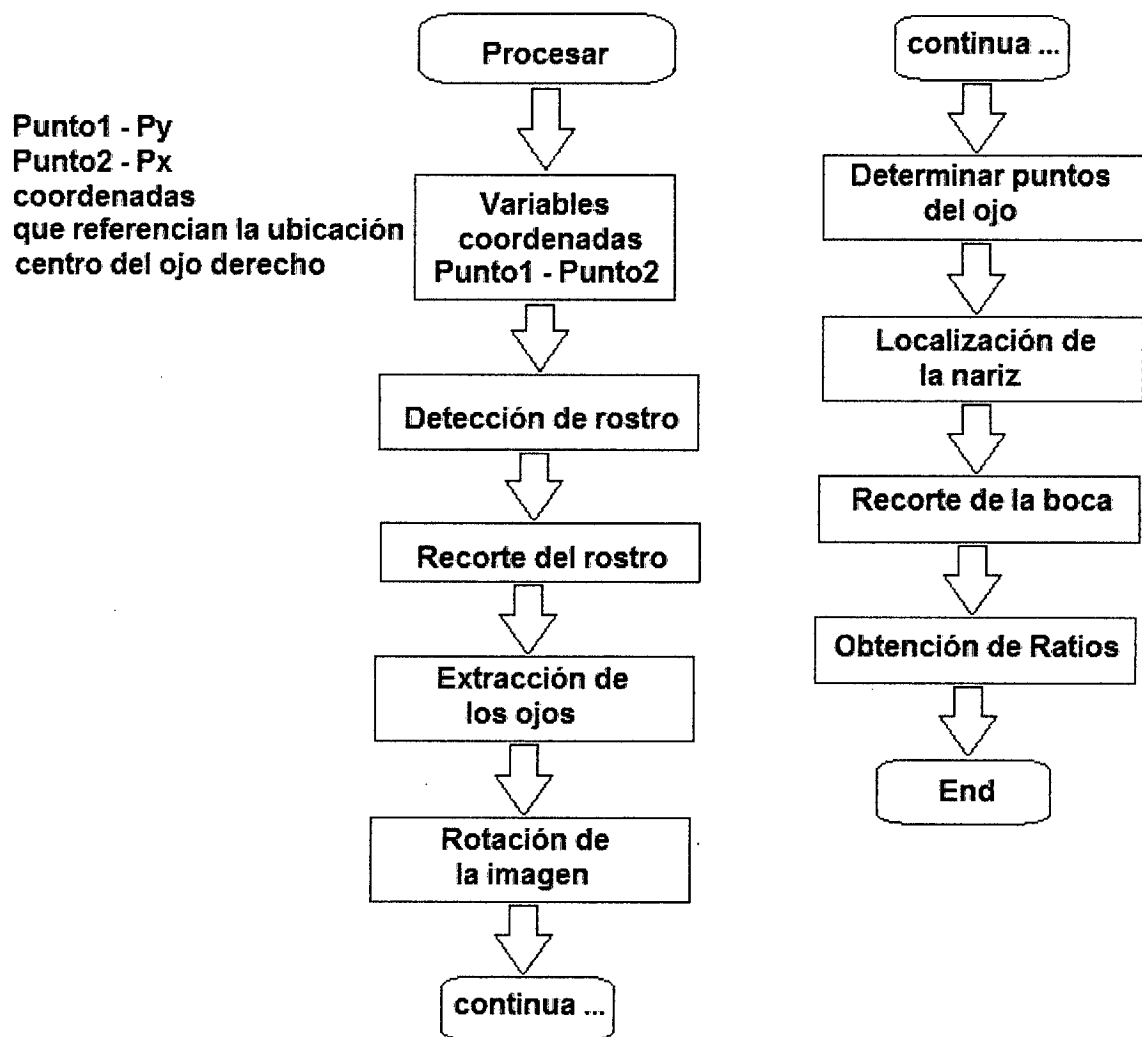


Fig. 3.9 Algoritmo de Reconocimiento PCA

A continuación se describe el código del programa desarrollado en una Interface Gráfica de Usuario GUI de Matlab. En cada caso se mostrará el procesamiento de la imagen para cada una de las etapas descritas en el Algoritmo que se muestra en la figura 3.9.

#### A. VARIABLES COORDENADAS.

Px y Py son las coordenadas que refencian los puntos de ubicación del centro del ojo derecho con respecto a una distancia tomada desde el ojo izquierdo para realizar las comparaciones necesarias y evaluar la similitud existentes entre la imagen que se procesa y las imágenes almacenadas en la base de datos.

## B. DETECCION DEL ROSTRO.

La imagen de entrada, que se capturo con la webcam y se guardo en la carpeta imágenes con el nombre de input.jpg, se almacena en la variable im0 y lo primero que se hace es extraerle la componente R ( $imR=im0(:,:,1)$ ) y convertirla a escala de grises ( $img=rgb2gray(im0)$ ). Paso seguido se resta de la variable imR que contiene la componente roja, la componente de escala de grises img. El resultado se almacena en la variable imG.



Fig. 3.10 Imágenes im0, imR, img e imG respectivamente

Después de obtener la imagen imG, esta es filtrada obteniendo una imagen más suave ( $imG=medfilt2(imG, [40 40])$ ) para luego darle mas brillo ( $imG=imadjust(imG)$ ) que permita convertirla a imagen binaria en base al calculo de un valor de umbral (entre 0 y 1)



Fig. 3.11 Imágenes para convertir a escala de grises

Paso seguido etiquetamos las zonas de color blanco con el propósito de elegir la zona de mayor area que representara el área de trabajo. Por ejemplo en la figura 3.12 (imagen binaria) se muestran cuatro áreas o zonas de color blanco. Esta imagen es etiquetada ( $bw = uint8(bwlabel(imB)))$  para obtener las propiedades de las áreas blancas como son coordenadas del centroide y límites del recuadro que encierra un área determinada como son coordenadas lado superior izquierdo (EjeBX, EjeBY), ancho (w) y largo (h).

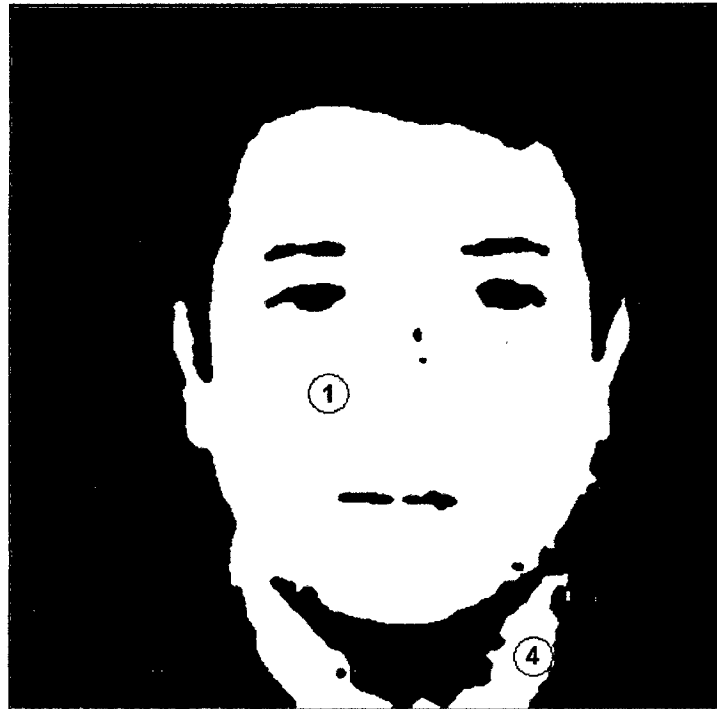


Fig. 3.12 Imagen para detectar áreas en blanco

Para obtener las propiedades de las regiones etiquetadas se usa el siguiente código:

`etq = regionprops(bw,'basic')` donde las propiedades básicas que se obtienen son: Área, Centroid y BoundingBox ya descritas líneas arriba. Al ejecutar dicha sentencia se obtiene la siguiente información para la etiqueta 1:

`etq(1)` Propiedades de la región etiquetada número 1

Area: 75667 Area = ancho (w) x largo (h)

Centroid: [335.8462 263.6961] Centroide [Ejex Ejey]

BoundingBox: [192.5 85.5 298 395] Sección rectangular de la región etiquetada número 1  
[EjeBx, EjeBY w h]

Por ejemplo, el ancho (w) de la sección rectangular que encierra el área etiquetada número 1 se da como:

`Etq(1).BoundingBox (3)`

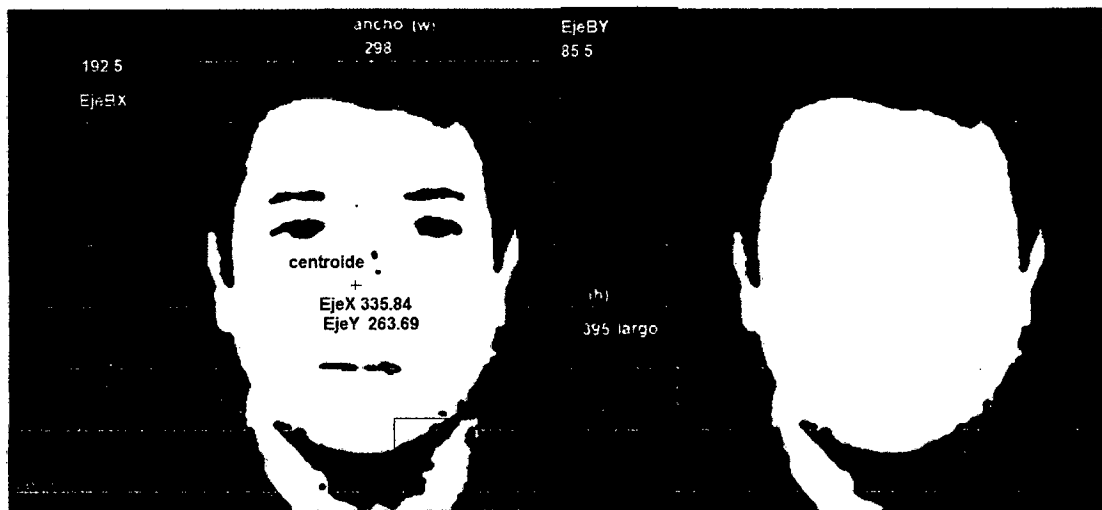


Figura 3.13 Características del Área etiquetada mayor

Estos datos corresponden al área máxima. Es decir, tal como se muestra en la figura 3.13, en la imagen existen cuatro áreas de color blanco, en donde el área que está encerrado en un recuadro verde, es el área máxima. Determinada el área máxima, se elimina el resto de áreas, además de asignar a cada pixel del área máxima un valor 1. A continuación se muestra el código para eliminar las áreas no necesarias:

```
Varea = zeros(1,N); % Vector de áreas de todas las regiones etiquetadas
for i = 1 : N
    Varea(i) = etq(i).Area;
end
[val ind1] = max(Varea); % El rostro es la máxima área etiquetada
clear Varea;

% val = valor de área máxima
% ind1 = indicador de área máxima

[x y] = find(bw~=0&bw~=ind1); % Eliminar otras áreas etiquetadas
for i = 1:length(x)
    imB(x(i),y(i))=0;
end
clear x y;

imB = imfill(imB,'holes'); % Se elimina "vacíos" dentro del rostro
                             para tener solo la plantilla del rostro
```

### C. RECORTE DEL ROSTRO.

Para determinar si el área de trabajo obtenida en la etapa de detección del rostro pertenece a un rostro humano se establece un primer criterio, el cual es que el ancho debe ser siempre menor que el largo. Esta información se extrae de la propiedad de área etiquetada:

```
bbx =etq(ind1).BoundingBox
```

es decir, para este caso,  $bbx = [192.5 \ 85.5 \ 298 \ 395]$  o  $pts = [192 \ 85 \ 298 \ 395]$ . La variable  $pts(3)$  corresponde al ancho del rostro y la variable  $pts(4)$  es el valor numérico de la altura del rostro. Esta información esta dada en pixels. Por lo tanto para determinar si estos parámetros corresponden a un rostro humano la división de  $pts(3)$  entre  $pts(4)$  debe ser menor que la unidad. Si esto es verdad, se procede a extraer información de área de trabajo así como la detección de los bordes de esta área.

```
bbx = etq(ind1).BoundingBox; % Obteniendo puntos del "Rectángulo" mayor
pts = floor(bbx);           % Convertir esos puntos a enteros >0

if pts(3)/pts(4)<1          % Condición de rostro humano

im1 = im0(pts(2):pts(2)+pts(4),pts(1):pts(1)+pts(3),:) % (a)
Bx = imB(pts(2):pts(2)+pts(4),pts(1):pts(1)+pts(3)); % (b)
Borde = bwperim(Bx);        % Se extrae el perímetro de la plantilla (c)

    sel = strel('line',3,0);
    se2 = strel('line',3,90);
Borde = imdilate(Borde,[sel se2]); % (d)
```

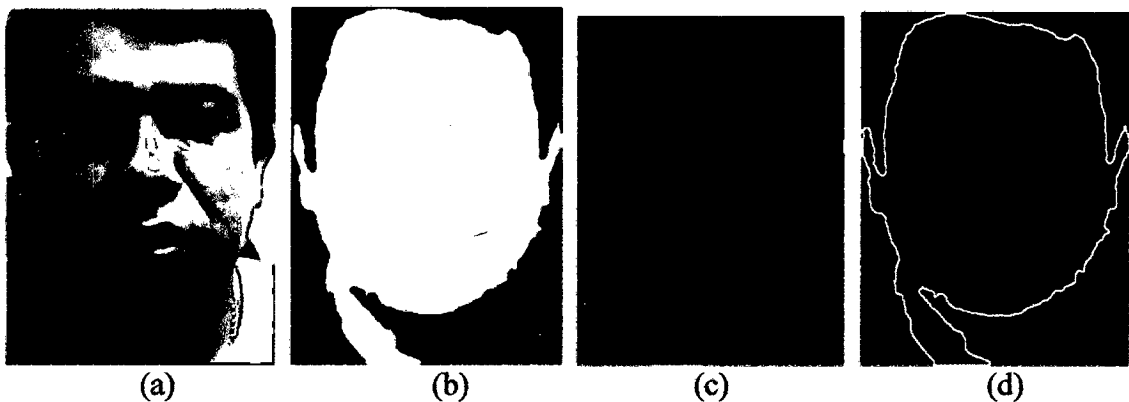


Fig. 3.14 Imágenes de detección de bordes del rostro

#### D. EXTRACCION DE LOS OJOS.

Se hace el mismo procedimiento para detectar el color carne pero con mayor sensibilidad a las variaciones de color. Se antepone condiciones para casos de no haber regiones etiquetadas dentro del rostro, una de las condiciones de los ojos su centro debe de estar en la mitad superior del rostro, la distancia horizontal debe ser mayor a la cuarta parte del ancho del rostro y la distancia vertical debe ser menor a la quinta parte de la altura del rostro.

```
imR = im1(:,:,1).*uint8(Bx); % Se hace el mismo procedimiento para
                                detectar el color carne pero con mayor
                                sensibilidad a las variaciones de color
```





Fig. 3.15 Detección color carne región del rostro

```
img = rgb2gray(im1).*uint8(Bx);           % (a)
imG = imsubtract(imR,img);               % (b)
imG = medfilt2(imG, [15 15]);            % (c)
imG = imadjust(imG);                     % (d)
uml = graythresh(imG);                    % umbral conversión a imagen binaria
imB = im2bw(imG,uml);                    % (e)
imBt = ~ (imB|Borde);                    % (f)
BW = bwlabel(imBt);
etq = regionprops(uint8(BW),'basic');
N = length(etq);
[fil col]=size(BW);                      % filas y columnas de la imagen
area = fil*col;
```

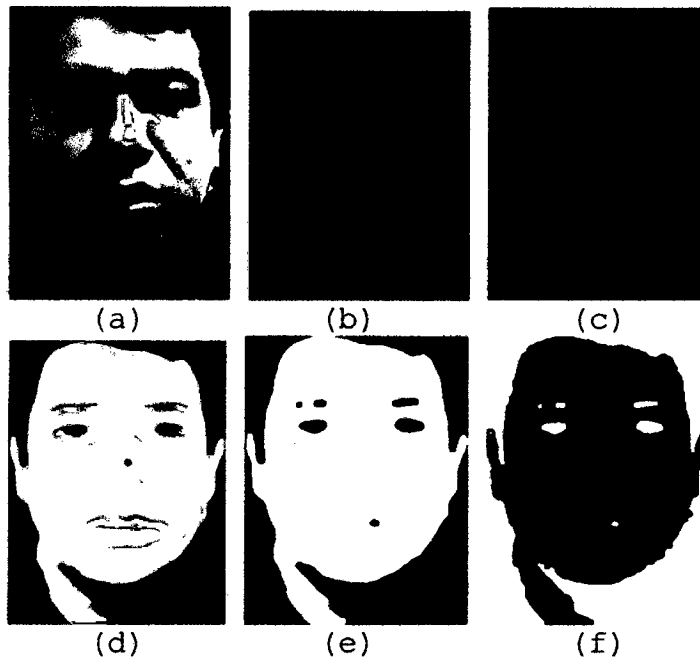


Fig. 3.16 Detección de los ojos

Para determinar la posición donde se encuentran los ojos del rostro se debe seguir las siguientes tres condiciones:

- ✓ El área para los ojos con relación al área de la imagen de rostro para la resolución RGB establecida debe estar en el rango de 0.1% a 2.5%.

- ✓ La posición donde se encuentra el centro de los ojos debe estar en la zona de la posición mitad superior de la imagen del rostro.
- ✓ La relación de los parámetros ancho (w)/altura (h) del área rectangular que encierra un ojo debe estar entre 1 a 3.5.

A continuación se tiene las líneas de código que permiten cumplir las condiciones mencionadas:

```
% se utiliza los siguientes vectores

Varea = zeros(1,N);           % Relación entre áreas de regiones
                              % etiquetas y el área del Rostro
Vctry = zeros(1,N);          % Centro de las regiones etiquetadas

Vrlbb = zeros(1,N);           % Relación ancho/altura de las
                              % regiones etiquetadas
% código para hallar los parámetros de todas las áreas etiquetadas

for i = 1:N
    Varea(i) = 100*etq(i).Area/area;
    Vctry(i) = etq(i).Centroid(2);
    Vrlbb(i) = etq(i).BoundingBox(3)/etq(i).BoundingBox(4);
end

% código para determinar los índices que cumplen las condiciones

indices1 = find(Varea>0.1 & Varea<2.5 & Vctry<fil/2 & Vrlbb>1 &
Vrlbb<3.6);
```

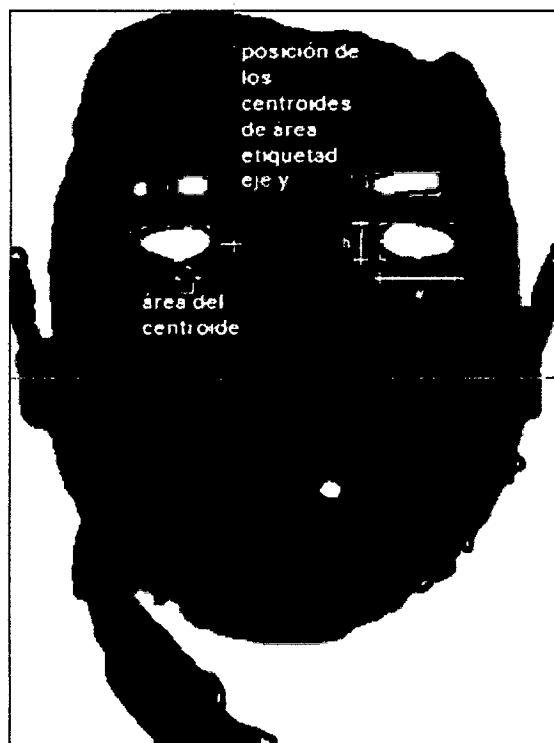


Fig. 3.17 Detección de áreas correspondiente al rostro

En la figura 3.17 se aprecian 19 áreas blancas, las cuales son etiquetadas con los parámetros de área, centroide y características del recuadro que las encierra como son las coordenadas del vértice superior izquierdo. Ya se mencionó que esto lo realiza el comando "regionprops" del Matlab. La información que se obtiene de la imagen de la figura 3.17, la cual es la tomada como muestra para describir la aplicación de reconocimiento de rostros; son cuatro áreas que cumplen con las condiciones establecidas y son las que corresponden a las encerradas en los rectángulos celestes. Estas áreas son las regiones etiquetadas indicadas en la variable "indices1" 7, 9, 14 y 15. Es por tanto que la condición de áreas es mayor a dos sabiendo que solo se evalúa como mínimo dos que son los ojos derecho é izquierdo. El siguiente paso es determinar las coordenadas que corresponden a las áreas etiquetas que cumplen con las tres condiciones mencionadas anteriormente. La finalidad es hallar la distancia que existe entre estas áreas para determinar que par de áreas cumplen con la condiciones siguientes denominada 4ta condición:

- ✓ La distancia Horizontal debe ser mayor a la 4ta parte del ancho del Rostro
- ✓ La distancia Vertical debe ser menor a la 5ta parte de la altura del Rostro



Fig. 3.18 Distancia horizontal y vertical entre los ojos

El código para realizar las condiciones de distancia es el siguiente:

```
% Coordenadas x - y de las áreas etiquetadas en el indice1

for i=1:length(indices1)
    Vctry(i) = etq(indices1(i)).Centroid(2);
    Vctrx(i) = etq(indices1(i)).Centroid(1);
end

% Ordenando los vectores según Vctry y Vctrx de mayor a menor

for i=1:length(Vctry)-1
    for j=i+1:length(Vctry)
        if Vctry(j)>Vctry(i)
            aux = Vctry(i);
            Vctry(i) = Vctry(j);
```

```

        Vctry(j) = aux;
        aux = Vctrx(i);
        Vctrx(i) = Vctrx(j);
        Vctrx(j) = aux;
        aux = indices1(i);
        indices1(i) = indices1(j);
        indices1(j) = aux;
    end
end

% Determinando se cumpla la 4ta condición

ojos_loc = 0;
for i=1:length(Vctry)-1
    aux = abs(Vctrx(i+1)-Vctrx(i));
    if aux>col/4
        aux = abs(Vctry(i+1)-Vctry(i));
        if aux<fil/5
            ojos_loc = 1;
            break; % En caso se cumpla la 4ta condición
        end
    end
end

% Ordenando Ojo1 y Ojo2 como Ojo Derecho e Izquierdo
if Vctrx(i)<Vctrx(i+1)
    ojo1(1) = round(Vctrx(i)); % CentroideX ojo derecho
    ojo1(2) = round(Vctry(i)); % CentroideY ojo derecho
    ojo2(1) = round(Vctrx(i+1)); % CentroideX ojo izquierdo
    ojo2(2) = round(Vctry(i+1)); % CentroideY ojo izquierdo
else
    ojo2(1) = round(Vctrx(i));
    ojo2(2) = round(Vctry(i));
    ojo1(1) = round(Vctrx(i+1));
    ojo1(2) = round(Vctry(i+1));
end
end

```

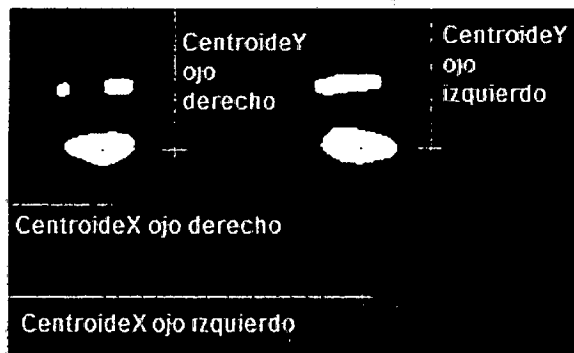


Fig. 3.19 Coordenadas de los centroides de los ojos

## E. RECORTE DE LA IMAGEN DE LOS OJOS

Para realizar el recorte de las imágenes de los ojos del rostro humano, se considerará que su forma será cuadrada, donde la dimensión del lado será la mitad de la distancia que separa ambos ojos. Se considerará además que las coordenadas deben ser mayor a cero y menores a las dimensiones del rostro.

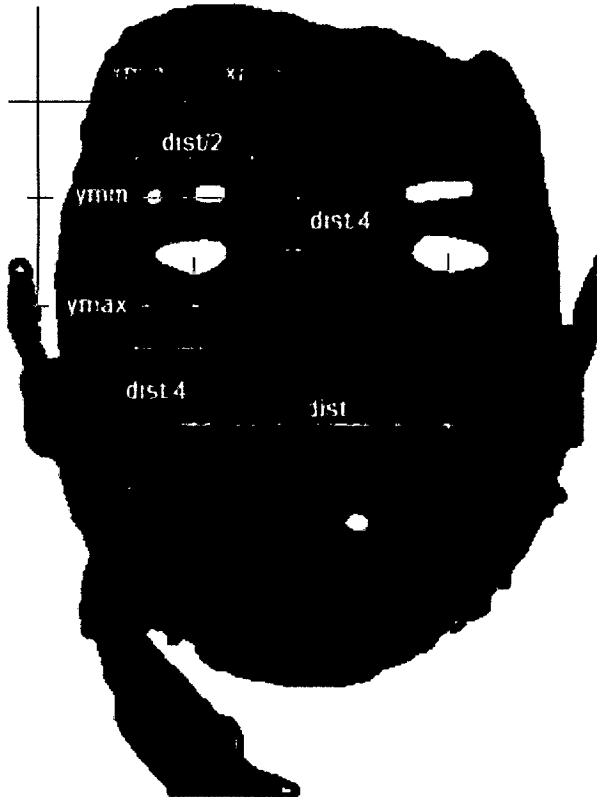


Fig. 3.20 Obtención de las imágenes de los ojos

% la variable `dist` se obtiene de restar el `CentroideX(2) - CentroideX(1)` y se divide entre cuatro(4), considerando que el área cuadrada que encierra la imagen de cada uno de los ojos tiene lado igual a  $dist/2$ .

```
dist = round(norm(ojo2-ojo1)/4);
```

% Obteniendo área cuadrada de la imagen del ojo derecho

```
if (ojo1(2)-dist)<1
    ymin = 1;
else
    ymin = ojo1(2)-dist;
end
if (ojo1(2)+dist)>size(im1,1)
    ymax = size(im1,1);
else
    ymax = ojo1(2)+dist;
```

```

end
if (ojo1(1)-dist)<1
    xmin = 1;
else
    xmin = ojo1(1)-dist;
end
if (ojo1(1)+dist)>size(im1,2)
    xmax = size(im1,2);
else
    xmax = ojo1(1)+dist;
end
OJO1 = im1(ymin:ymax,xmin:xmax,:); % Imagen ojo derecho

% Obteniendo área cuadrada de la imagen del ojo izquierdo

if (ojo2(2)-dist)<1
    ymin = 1;
else
    ymin = ojo2(2)-dist;
end
if (ojo2(2)+dist)>size(im1,1)
    ymax = size(im1,1);
else
    ymax = ojo2(2)+dist;
end
if (ojo2(1)-dist)<1
    xmin = 1;
else
    xmin = ojo2(1)-dist;
end
if (ojo2(1)+dist)>size(im1,2)
    xmax = size(im1,2);
else
    xmax = ojo2(1)+dist;
end
OJO2 = im1(ymin:ymax,xmin:xmax,:); % Imagen ojo izquierdo

```



Fig. 3.21 Ojo derecho y ojo izquierdo

El siguiente paso es la ubicación del centro de cada uno de los ojos. Para esto, se ubicará la pupila para que cumpla esa función debido a que por cuestiones de luz, la parte de las cejas y de los ojos pueden formar una sola región binaria. El código se describe a continuación:

```
% las variables o1(ojo derecho) y o2(ojo izquierdo) contienen las
coordenadas de los centroides de cada uno de los ojos

o1 = ojo1;
o2 = ojo2;

img = rgb2gray(OJO1); % imagen de los ojos a escala de grises

img = histeq(img); % Ecualización de la imagen de los ojos
                  % Suavizado de la imagen para llevar a una
luminosidad "uniforme"
```



**Fig. 3.22 Imágenes ecualizadas**

```
b = 20;
a = (255-b)/log(255+1);
img = a*log(double(img)+1)+b;
img = uint8(img);
um1 = graythresh(img);
imb = im2bw(img,um1); % Binarización de las imágenes
bw = bwhlabel(~imb);
```



**Fig. 3.23 Imágenes binarizadas**

```
etq = regionprops(uint8(bw),'basic'); % Etiquetar regiones negras
N = length(etq);
```

Ahora se procede a determinar condiciones que corresponden al área de las pupilas con respecto al área total del rostro y su ubicación considerando que las cejas están en la parte superior del ojo

```
Vctry = zeros(1,N);
[fil col] = size(bw);

% Condiciones de las pupilas:
% Área > 5%
% Posición: parte baja del OJO
% Forma: ancho>=altura
for j=1:N
    if etq(j).BoundingBox(3)/etq(j).BoundingBox(4)>1 &&
        100*etq(j).Area/(fil*col)>5
        Vctry(j)=etq(j).Centroid(2);
    end
```

```
end
[val ind1] = max(Vctry);
```

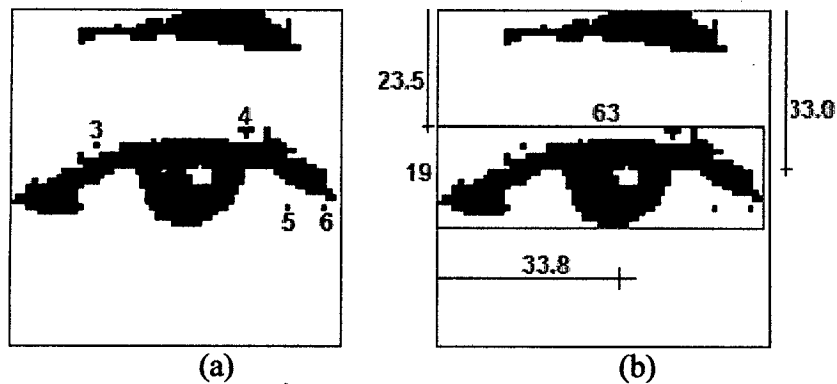


Fig. 3.24 Áreas etiquetadas del ojo derecho

En la figura 3.24(a) se muestra el área de la imagen binaria que corresponde al ojo derecho. Ahí se aprecian seis áreas donde el área uno (1) corresponde a la que encierra la pupila del ojo derecho la que se determinará luego, como el centro del ojo derecho para realizar el cálculo de la distancia hacia el centro del ojo izquierdo. El comando en Matlab `etq(1)`, que corresponde al área etiquetada número 1; muestra la siguiente información:

```
>> etq(1)
ans =
    Area: 510
  Centroid: [33.8275 33.0039]
 BoundingBox: [1.5000 23.5000 63 19]
```

Como vemos, esta información está referida a los parámetros que se muestran en la figura 3.24(b).

## F. ROTACION DE LA IMAGEN DEL ROSTRO.

En esta parte se procederá a rotar la imagen del rostro de tal manera que la distancia entre los Centroides Y de los ojos sea cero. La figura 3.19 hace referencia a los parámetros que permiten el cálculo del ángulo de rotación aplicando distancia de ordenadas entre distancia de abscisas.

```
% Calculo del ángulo de rotación ang

aux = (ojo2(2)-ojo1(2))/(ojo2(1)-ojo1(1));
ang = atand(aux);
msc = zeros(size(BW));
msc(ojo1(2),ojo1(1)) = 255;
msc(ojo2(2),ojo2(1)) = 255;
mscr = imrotate(msc,ang);
```



```

[y x]=find(mscr==255);
ojo1 = [min(x),max(y)];
ojo2 = [max(x),max(y)];
F = max(x)-min(x);
im1 = imrotate(im1,ang)

```

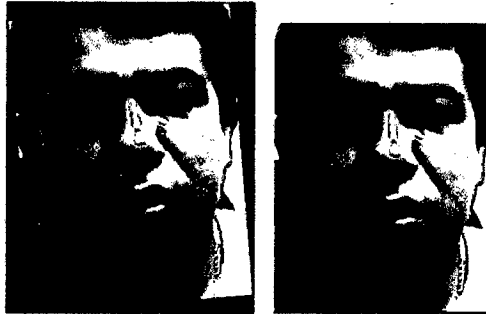


Fig. 3.25 Rotación de imagen del rostro

Ahora se procede a calcular la distancia entre los centros de los ojos a los cuales se le asigna la variable F. Conociendo los centroides de las imágenes de los dos ojos, se crea una mascara de la imagen donde solo se asigna el color blanco (255) a las posiciones de los centroides de los ojos tal como se muestra en el siguiente código.

```

msc = zeros(size(BW));
msc(ojo1(2),ojo1(1)) = 255;
msc(ojo2(2),ojo2(1)) = 255;

```

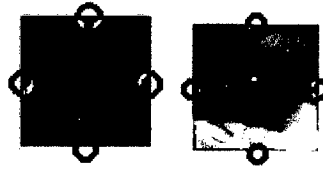
#### **G. DETERMINACION DE LOS DEMAS PUNTOS DEL OJO.**

Esta operación se realiza en las imágenes de los ojos ya rotados ahora se procede a calcular la distancia entre los centros de los ojos a los cuales se le asigna la variable F. Conociendo los centroides de las imágenes de los dos ojos, se crea una mascara de la imagen donde solo se asigna el color blanco (255) a las posiciones de los centroides de los ojos tal como se muestra en el siguiente código

```

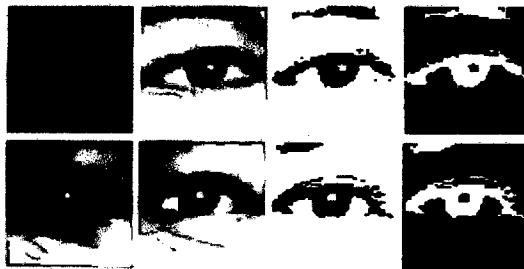
OJO1 = im1(ojo1(2)-round(F/4):ojo1(2)+round(F/4),ojo1(1)-
round(F/4):ojo1(1)+round(F/4),:);
OJO2 = im1(ojo2(2)-round(F/4):ojo2(2)+round(F/4),ojo2(1)-
round(F/4):ojo2(1)+round(F/4),:);

```



```
for i = 1:2
```

```
    if i==1
        img = rgb2gray(OJO1);
    else
        img = rgb2gray(OJO2);
    end
    img = histeq(img);
    b = 50;
    a = (255-b)/log(255+1);
    img = a*log(double(img)+1)+b;
    img = uint8(img);
    um1 = graythresh(img);
    imb = im2bw(img,um1);
    bw = bwlabel(~imb);
```



```
    etq = regionprops(uint8(bw),'basic');
    N = length(etq);
    if N~=0
        clear Vctrx Vctry
        for j=1:N
            Vctrx(j) = etq(j).Centroid(1);
        end
        if i==1
            [val ind1] = min(Vctrx);
            ojo1(3) = ojo1(1)-
round(F/4)+round(etq(ind1).BoundingBox(1));
            ojo1(4) = ojo1(2);
            ojo1(5) = ojo1(1)+round(F/4);
            ojo1(6) = ojo1(2);
            ojo1(1) = round((ojo1(3)+ojo1(5))/2);
        else
            [val ind1] = max(Vctrx);
            ojo2(3) = ojo2(1)-round(F/4)+1;
            ojo2(4) = ojo1(2);
            ojo2(5) =
ojo2(3)+round(etq(ind1).BoundingBox(1)+etq(ind1).BoundingBox(3)-1);
```

```

                                ojo2(6) = ojo1(2);
                                ojo2(1) = round((ojo2(3)+ojo2(5))/2);
                                end
                                else
                                disp(['No ubicados puntos 2 y 3 de OJOS -> '
num2str(i)])
                                end
                                end
                                end

```

Para el reconocimiento de la nariz y la boca, se sigue los mismos criterios de identificación vistos anteriormente y que están comentados en el código del programa.

## **H. OBTENIENDO LOS RATIOS.**

- **RELACION DE LAS DISTANCIAS ENTRE OJOS, NARIZ Y BOCA**

```

R(1) = F;
R(2) = abs(labio1(1)-labio2(1));
R(3) = abs(ojo1(2)-labio1(2));
R(4) = abs(ojo2(2)-labio2(2));
R(5) = abs(ojo1(2)-nariz3(2));
ind1 = 1;
r = zeros(1,10);
for i=1:4
for j=i+1:5
r(ind1) = R(i)/R(j);
ind1 = ind1+1;
end
end
end

```

# **CAPITULO IV**

# **PRUEBAS Y**

# **CONCLUSIONES**

**TESIS PARA OBTENER EL TITULO PROFESIONAL  
INGENIERO ELECTRONICO Y TELECOMUNICACIONES  
UNIVERSIDAD NACIONAL DE PIURA**

## 4.1 PRUEBAS REALIZADAS

El programa de detección de rostros empleando el método PCA, permite identificar a un individuo siempre y cuando sus datos tipo imagen JPG y sus características denominadas patrones del rostro están almacenados en una base de datos, en este caso un archivo XLS. A continuación, se muestran las pruebas realizadas al programa de reconocimiento de donde se obtuvieron observaciones que permitieron la elaboración de las conclusiones del proyecto de investigación.

### 4.1.1 VALIDEZ DE IMAGEN DE ROSTRO HUMANO

El programa de identificación lo primero que hace es obtener una máscara equivalente al rostro humano en la cual están establecidas los patrones de acuerdo al ancho y alto de acuerdo al punto B (detección del rostro) para el procesamiento de archivo input.jpg visto en el ítem 3.3.3.4 del capítulo III de la tesis. Si la imagen no corresponde a un rostro humano, se muestra un mensaje tal como se ve en la figura 4.1

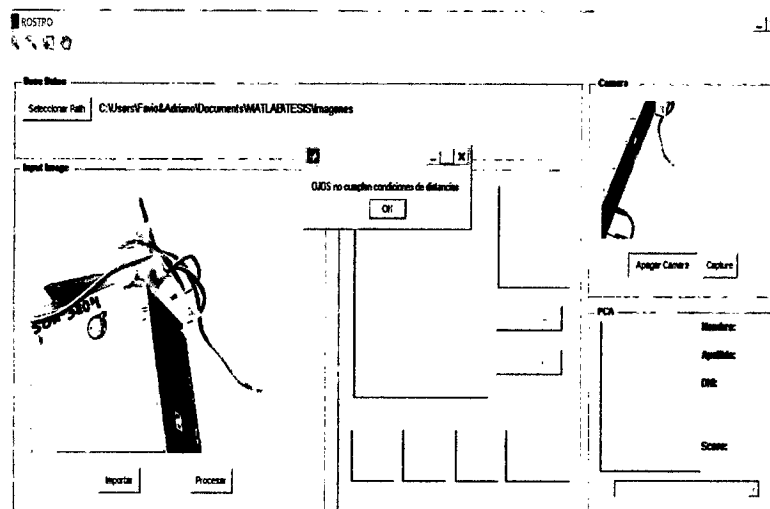


Fig. 4.1 Reconocimiento de imagen de rostro no valida

En la figura 4.2 se puede observar que la imagen capturada no cumple con los parámetros de etiquetado para la máscara correspondiente a un rostro, sobre todo al ser analizado de acuerdo a la existencia de áreas que correspondería a los ojos tal como se vio en el punto E de recorte de la imagen de los ojos (ítem 3.3.3.4 del capítulo 3)

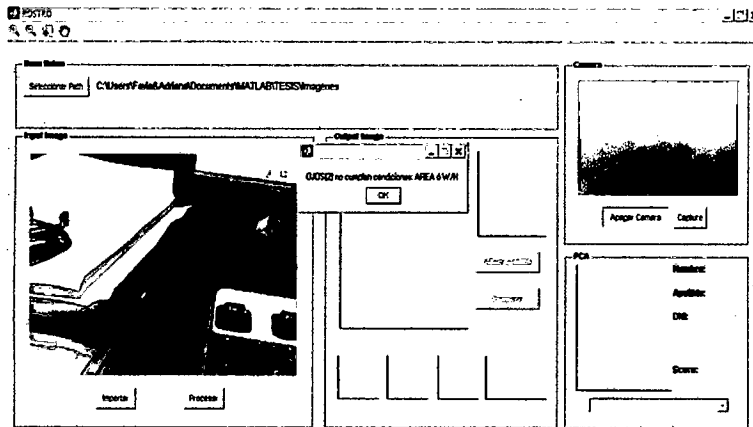
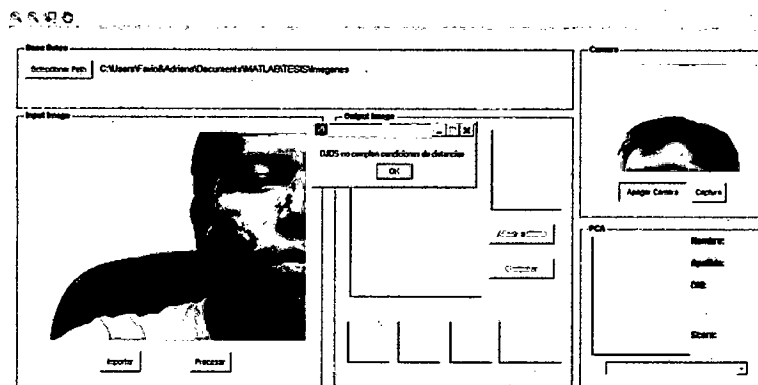


Fig. 4.2 Reconocimiento no válido para el etiquetado de las áreas de los ojos.

#### 4.1.2 IMAGEN DE ROSTRO INCOMPLETO.

La figura 4.3 muestra una imagen de rostro que no cumple con parámetros establecidos, nos estamos refiriendo a los patrones de distancia entre ojos, al no encontrar referenciado el punto1 y el punto2, la secuencia establecida para seguir la ruta se interrumpe.



En la fig. 4.3 Distancia entre ojos no cumple patrones

#### 4.1.3 AREA DE LA MASCARA DE ROSTRO NO VALIDA

Al igual que en la figura 4.2, al estar el rostro incompleto tal como se muestra en la figura 4.4, no se cumplen los patrones de etiquetado del área del rostro

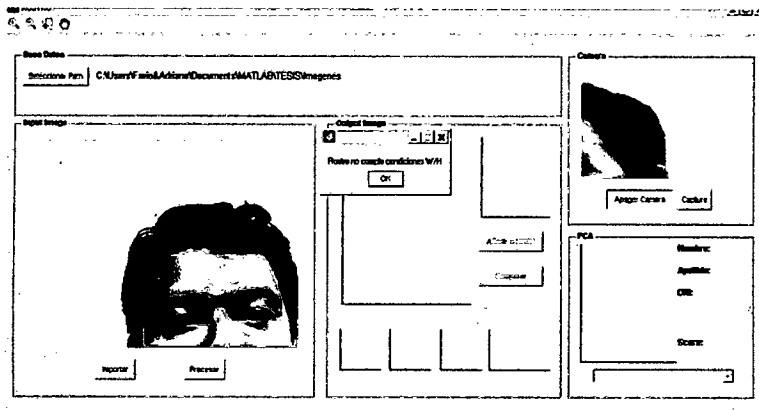


Fig. 4.4 Etiquetado del rostro no valido

#### 4.1.4 DISTANCIA MEDIDA A 70 CM.

Para distancias entre intervalos de 20 a 70 centímetros del rostro – cámara, el sistema reconoce siempre y cuando la cámara enfoque la totalidad del rostro.

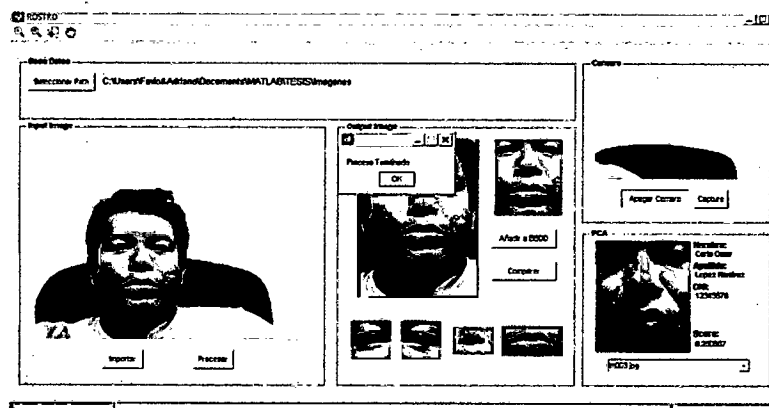


Fig. 4.5 Identificación de rostro válido

## 4.2 CONCLUSIONES EN BASE OBJETIVOS ESPECIFICOS PROPUESTOS.

- Se diseño e implemento una aplicación de reconocimiento de rostros usando el algoritmo de reconocimiento biométrico PCA utilizando las herramientas para el procesamiento de imágenes de Matlab.
- El software desarrollado en Matlab integra los módulos de captura, procesamiento, almacenamiento e interfaz para el sistema de reconocimiento de rostros.

- Para la captura de imágenes para lo cual se utilizó una webcam MICRONICS PLUG&PLAY de 8MP y el algoritmo de detección y configuración visto en el ítem 3.3.3.2 del capítulo 3.
- El procesamiento de la imagen para el reconocimiento de rostros se realizó con el uso del algoritmo PCA o análisis de componentes principales, el cual consiste en determinar las zonas del rostro, etiquetarlas y determinar mediante parámetros de área, ancho y alto si corresponden al rostro humano.
- Las imágenes procesadas se almacenaron con la extensión tipo JPG y sus parámetros de comparación fueron almacenados en un archivo XLS.
- La interfaz de usuario se realizó con la herramienta GUIDE del Matlab.
- Se estudió distintas técnicas de procesamiento digital de imágenes como PCA, Reconocimiento facial 2D, el método EGBP, entre otros; con la finalidad que me permita elegir la más apropiada para la aplicación.

## **CONTRASTACION DE LA HIPÓTESIS**

*El conocimiento de herramientas de desarrollo para el procesamiento de imágenes de Matlab permitió el desarrollo del algoritmo se usaron herramientas de captación de imágenes, configuración de cámara webcam, interfaz guide, escala de grises, sustracción de componentes, filtros, ajustes, umbrales, etiquetado de imágenes, región pros, etc. Los algoritmos de reconocimiento biométrico fueron diseñados con referencia a la técnica PCA (Análisis de Componentes Principales) dieron origen al diseño e implementación de una aplicación para el reconocimiento de rostros humanos*



# **BIBLIOGRAFIA**

## **REFERENCIAS**

**INGENIERIA ELECTRONICA  
Y TELECOMUNICACIONES  
TESIS DE GRADO PARA OPTAR EL TITULO PROFESIONAL**

1. Face Recognition Software:

En [http://centrodeartigos.com/articulos-informativos/article\\_75905.html](http://centrodeartigos.com/articulos-informativos/article_75905.html)

2. "Análisis comparativo de métodos basados en subespecies aplicados al reconocimiento de caras." En <http://www.uv.es/marjoari/pdf/definitivo.pdf>
3. Henry Arguello Fuentes, Sistemas de reconocimiento basados en imágenes Faciales, Colombia, Publicado junio 2011, Revisado Diciembre del 2013, disponible en: <http://www.bdigital.unal.edu.co/22505/1/19127-106814-1-PB.pdf>.
4. D. Carrero, B. Ruiz, L. Puente, M.J. Poza, Prestaciones de la normalización del rostro en el reconocimiento Facial, España, Revisado Febrero del 2014, disponible en: <http://jrpb10.unizar.es/papers/S5.C3.pdf>.
5. Carmen Virginia Gámez Jiménez, Diseño y desarrollo de un sistema de reconocimiento de caras, España, Publicado Abril 2009, revisado Febrero del 2014, disponible en: [http://e-archivo.uc3m.es/bitstream/handle/10016/5831/PFC\\_CarmenVirginia\\_Gamez\\_Jime nez.pdf?sequence=1](http://e-archivo.uc3m.es/bitstream/handle/10016/5831/PFC_CarmenVirginia_Gamez_Jime nez.pdf?sequence=1).
6. Roger Jimeno Hernández, Estudio de técnicas de reconocimiento Facial, España, Publicado Mayo 2010, Revisado en enero del 2014, disponible en: [http://upcommons.upc.edu/pfc/bitstream/2099.1/9782/1/PFC\\_RogerGimeno.pdf](http://upcommons.upc.edu/pfc/bitstream/2099.1/9782/1/PFC_RogerGimeno.pdf).
7. Mauricio Delbracio, Matias Mateu, Identificación utilizando PCA, LDA, Publicado Marzo del 2006, Revisado en enero del 2014, disponible en: [http://iie.fing.edu.uy/investigacion/grupos/biometria/proyectos/patrones/RecPat\\_M M.pdf](http://iie.fing.edu.uy/investigacion/grupos/biometria/proyectos/patrones/RecPat_M M.pdf).